



**DOKUZ EYLÜL ÜNİVERSİTESİ**  
**İZMİR MESLEK YÜKSEKOKULU**  
**BİLGİSAYAR TEKNOLOJİLERİ BÖLÜMÜ**

# **DMA DATA MANAGEMENT APPLICATION**

Umut Güngör 2021730029

İzmir  
Mayıs 2023

**Tez Danışmanı**  
**Öğr. Gör. İsmail DÜŞMEZ**

**PART I**

# İçindekiler

Onay .....	3
Etik İlkelere Uygunluk Beyanı.....	4
Teşekkür .....	5
Özet.....	<b>Hata! Yer işareti tanımlanmamış.</b>
1.proje fikrinin bilgisinin bulunması.....	7
2.araştırma süreci tasarlanması (05.03.2023) .....	7
3.zaman planlaması .....	7
4.toplantı kararları .....	10
5.Analizler .....	13
6.tasarım .....	17
7.kodlama .....	17
PART I Bitiş .....	115
1.Tasarım için kütüphane arayışı .....	118
2.Programların Kurulması .....	122
3.Tasarım Fikirlerinin Tartışılması .....	123
4. Kütüphaneleri Öğrenme .....	124
5. HTML Sayfalarının Tasarlanması .....	125
6.HTML kodları.....	140
7.Yönlendirmeler ve Etkileşimler (TS kodları) .....	195
8.CSS Kodları .....	303
9.Kod Açıklamaları .....	306
10.test ve hataların bulunması. ....	308
11.proje paketleme ve teslim. ....	309

## Onay

Umut G ng r tarafından hazırlanan **DMA Data Management Application** bařlıklı bitirme projesi, kapsam ve nitelik aısından kalite standartlarına uygunluęu ile ilgili Dokuz Eyl l  niversitesi İzmir Meslek Y ksekokul Bilgisayar Teknolojileri b l m  olarak 02/0//2023 tarihinde kabul edilmiřtir.

Proje Y r t c s 

Adı – Soyadı

İmza

Danıřman

 ğr. G r. İsmail D řMEZ

05/06/2023

## **Etik İlkelerle Uygunluk Beyanı**

Bu projenin içinde sunduğum verileri, bilgileri ve belgeleri akademik ve etik kurallar çerçevesinde elde ettiğimi; tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu; çalışmada bana ait olmayan tüm veri, düşünce, sonuç ve bilgilere bilimsel etik kurallar gereği olarak eksiksiz şekilde uygun atıf yaptığımı ve kaynak göstererek belirttiğimi beyan ederim.

**Öğrenci Adı ve Soyadı**

**Umut Güngör**

**01/06/2023**

## **Teşekkür**

Bu bitirme projesi çalışmasında desteklerinden dolayı Hidayet Doğru ve Levent Yusufoglu'na teşekkür ederim.

**Öğrenci Adı ve Soyadı**

**Umut Güngör**

## **Özet**

**Umut Güngör**

**Ön Lisans Bitirme Projesi, Bilgisayar Bilim Dalı Mayıs, 2023 303 sayfa**

Web tabanlı depo kontrol sistemi, bir işletmenin depolama süreçlerini etkin bir şekilde yönetmesini sağlayan bir yazılım sistemidir. Bu sistemin amacı, depo envanterini izlemek, stok seviyelerini kontrol etmek, mal kabul ve sevkiyat süreçlerini yönetmek ve depo faaliyetlerini optimize etmektir.

Web tabanlı depo kontrol sistemi, kullanıcı dostu bir arayüz üzerinden depo yöneticilerine, depo çalışanlarına ve diğer ilgili personellere depo operasyonlarını kolayca yönetme imkanı sunar.

Bu sistem stok seviyelerini izler. Depo çalışanları, sisteme erişerek stok seviyelerini kontrol edebilir ve stok durumuna bağlı olarak siparişleri hazırlayabilir. Bu, depoda stok eksikliği veya fazlalığının önlenmesine yardımcı olur ve işletmenin operasyonel verimliliğini artırır.

Web tabanlı depo kontrol sistemi, depo envanterini gerçek zamanlı olarak izleyerek envanter kayıplarını ve hatalarını azaltır. Sistem, her bir ürünün depoda bulunduğu konumu takip eder ve personelin ürünleri hızlı bir şekilde bulmasını sağlar. Ayrıca, depo operasyonlarının performansını analiz etmek ve iyileştirmek için veri ve raporlama özellikleri sunar.

Bu dokümantasyon, web tabanlı depo kontrol sisteminin kurulumu, yapılandırması ve kullanımıyla ilgili kapsamlı bir rehber sunmaktadır. Kullanıcılar, sistem gereksinimleri, kullanıcı hesapları oluşturma, ürün ve stok yönetimi, raporlama ve diğer ilgili konular hakkında ayrıntılı bilgilere erişebilirler.

Web tabanlı depo kontrol sistemi, işletmelere depo operasyonlarını otomatikleştirme, hataları azaltma ve verimliliği artırma imkanı sunar. Bu sistem, depo yöneticilerine ve çalışanlara envanterin daha iyi kontrolünü sağlar ve işletmelere rekabet avantajı kazandırır.

**Anahtar kelimeler:** Depo, kontrol, API, stok, web

## 1.proje fikrinin bilgisinin bulunması

Depo kontrol sistemi kullanırken nasıl yazıldığını merak etmek ile başladı göz ile görülebilir bazı eksikliklerin olduğunu düşünmek ise bu projeye başlama sebebini vermiş oldu. böyle bir sistemi yaparken ileride dönüp bu projeye baktığımızda bizim için bir rota olabileceğini düşünmek de kararı kesinleştirdi.

## 2.araştırma süreci tasarlanması (05.03.2023)

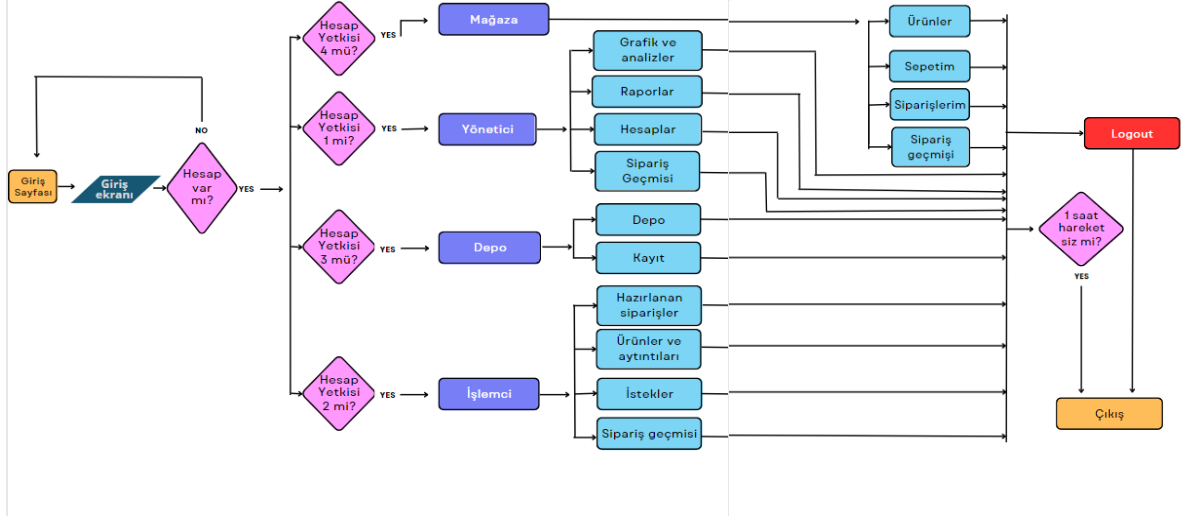
- Tasarım için benzer programların analizi.
- Depo kontrol sistemlerinin işleyişinin araştırılması.
- Kullanılacak programların araştırılması

## 3.zaman planlaması

### Toplantı Akış Şeması

TOPLANTILAR	04.03	19.03	31.03	15.04	30.04	07.05	15.05	26.05
PROJENİN AÇIKLANMASI VE FIKIR ALISVERİSİ								
PROJE GEREKSİNİM ANALİZİ VE GENEL İSLEYİS								
PROJEDE KULLANILAN PROGRAMLARIN TARTISILMASI								
PROJE TASARIMI İLE İLGİLİ TARTISILMASI								
WEB API TARAFINDA KULLANILCAK ALGORITMA								
PROJENİN İCİNDEKİ UYGULAMALARIN ENTEGRASYONU								
OLUSABİCELEK SORUNLAR HAKKINDA KONUSMA								
TEST VE SON AYARLAR								

### Akış Diyagramı





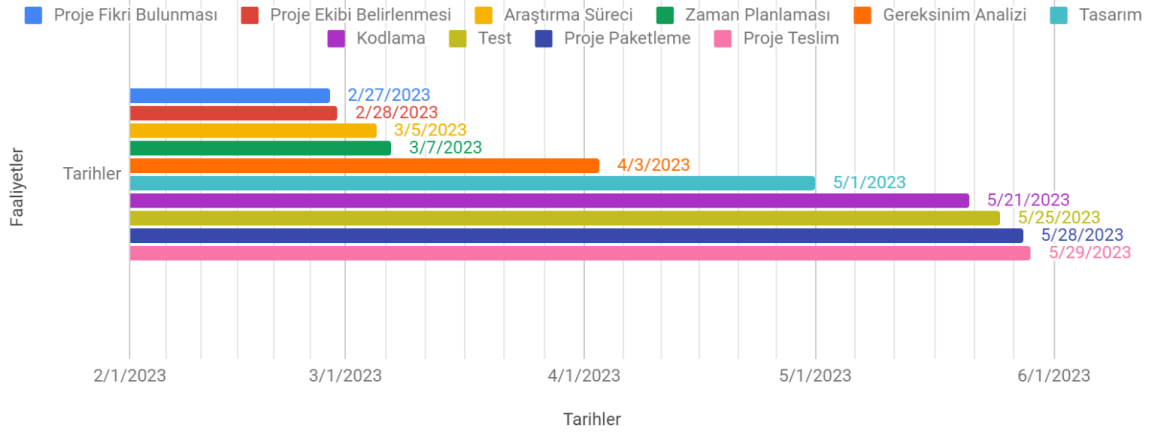
## Aytaç Melih Kılıç İş Akış Şeması

[illegible]

## Umut Güngör İş Akış Şeması

[illegible]

## Zaman Çizelgesi



**Not:** Ay/gün/yıl şeklinde tasarlanmıştır.

## 4.toplantı kararları

### İ. Projenin açıklanması ve fikir alışverişi:

Projeyi bu sektörde olan kişilere danışık fikirlerini belirtmelerini istedik yeni belirlediğimiz fikirlerin ne doğrultuda doğru olup olmadığını tartmasını istedik bizi doğru kaynaklara yönlendirerek daha hızlı sonuç almamızı sağladı.

### İİ. Toplantıda konuşulan konuların içerikleri:

- 1- Projenin hangi platformlarda kullanılabilir olması.
- 2- Projede kullanılan sistem hakkında bilgi alışverişi.
- 3- Projenin kapsamı.
- 4- Projenin amacı.
- 5- Projenin hedefi.

### Toplantıda alınan kararlar;

- 1- Projenin tüm platformlarda kullanılabilir olması için responsive bir tasarım yapılacak ve web tabanlı bir uygulama olmasına karar verilmiştir.
- 2- Depo kontrol sisteminde temel kontrolcülerin olması gerektiğini öğrendik böylelikle işlemci, depo görevlisi, mağaza ve yönetici ekranı yazmaya karar verdik.
- 3- işletme sahipleri, satın alma yöneticileri ve depo yöneticileri gibi stok yönetiminden sorumlu olan herkes bu sistemi kullanabilir. Bu sistem büyük ölçekli işletmeler için uygundur. İster tek bir depo isterse birden fazla depo ile çalışılsın, sistem tüm işletmelerin stok yönetim ihtiyaçlarını karşılamak için tasarlanmıştır. Ayrıca, bu sistem müşterilerin ihtiyaçlarına göre özelleştirilebilir olduğu için, farklı sektörlerdeki işletmelerin de kullanabileceği bir yapıdadır.
- 4- Firmaların stokları hakkında daha doğru ve detaylı bilgiye sahip olmaları, stok yönetimlerini daha etkili bir şekilde yapmaları, stoklarında meydana gelebilecek problemleri önceden tespit edebilmeleri ve dolayısıyla iş süreçlerini daha verimli hale getirebilmeleri hedeflenmektedir.

- 5- Stok kontrolü yapan firmaların ihtiyaçlarına yönelik bir web tabanlı stok kontrol sistemi geliştirmek ve bu sistemi kullanıma sunarak müşteri memnuniyetini artırmak, iş süreçlerini optimize etmek, hata payını minimuma indirmek ve firmaların rekabet gücünü artırmaktır. Ayrıca, sistemde yer alan veri analizi ve raporlama özellikleri sayesinde, firmaların stoklarına ilişkin detaylı istatistiksel verileri elde etmeleri ve stok yönetimlerini daha verimli bir şekilde yapmaları hedeflenmektedir.

### İİİ. Projede kullanılan programların tartışılması.

1-sistemin senkron-asenkron olaylarının hangisinin kullanılması gerektiğinin tartışılması.

2-tasarım için hangi kütüphanelerin kullanılması gerektiğinin tartışılması.

3- veri işleme işlemlerinin nereden yapılması gerektiği tartışılması.

4- hangi veri tabanı yönetim uygulaması kullanılacağı tartışılması.

Toplantı sonucu alınan kararlar:

- 1- Sistemlerin geç yüklendiğinden dolayı sistemin asenkron olarak çalışmasını istedik yani ekranda gözüken kısımlar sadece program çalıştığıda ekrana getirmek için çağrılıp daha az yoğunluk oluşması sağlanması gerektiği konuşulmuştur bundan dolayı bir freamework konusunda anguların kullanılması netlik kazanmıştır.
- 2- SCSS kütüphanesinin kullanılma amaçları, CSS yazımını daha modüler ve düzenli hale getirerek geliştirme sürecini kolaylaştırması, sayfaları daha iyi organize edebilir ve stil kurallarını daha etkili bir şekilde yönetilmesini kolaylaştırır, değişkenler kullanarak tekrar kullanılabilirlik sağlayabilir ve stilinizi daha etkili bir şekilde yönetir.

JSCHART kütüphanesinin kullanılma amaçları, grafik şemalarını oluşturması, veri gruplarından gelen veriler ile entegrasyonu olması ve otomatik olarak veriye göre kendini güncellemesi, kütüphanenin js ile uyumlu olması, şemaların farklı şekilde kullanıcıya sunulabilmesi, şemaların düzenli ve temiz bir görüntü sunması, responsive bir tasarım sunması, kullanımı kolay ve anlaşılabilir olması.

Bootstrap kütüphanesinin kullanılma amaçları, hazır şemalarla ve sistemleri ile tasarımın kolayca tasarlanmasını sağlması, düzenli şık bir görüntüye sahip olması, sürekli olarak güncel kalması, responsive bir yapıya sahip olması, kullanıcının alışkın olduğu bir yapıya sahip olması, grid,row,column,flex gibi sistemler bulundurması ve bu sistemleri hazır bir şekilde sunması.

- 3- Verilerin doğrudan projeye eklemek yerine daha güvenli ve daha kullanışlı bir şekilde kullanmak istediğimizden verileri veritabanından api kullanarak sisteme gönderdik böylelikle oluşabilecek veri kayıplarını engelledik üstelik farklı kullanıcıların kendi veri tabanları var ise bu kişileri apiyle beraber projeye entegre etmek daha kolaylaşır.
- 4- Sektörde sıklıkla kullanılan ve bizlerin de hakim olarak kullanabildiği sql server managment studio yu kullanmak daha faydalı gördük içerisindeki bir çok imkan ise bizlere daha fazla katkıda bulunacağını düşündük.

#### İV. Proje tasarımı konusunda tartışılması:

- 1- Renklerin uyumlu olmaması ve belirli kurallara uygun yapılması gerektiği tartışılması.

##### Alınan kararlar:

- 1- renklerin belirli oranlar ile site içerisinde tekrar kullanılması gerektiği ve fazla baskın olmaması gerektiği ve renk skalasının değişmesi renkler şöyledir #1d4b79, #ff8b07, #1ABC9C, #03A9F4, #FEFC107.

#### V. web API tarafında kullanılacak algoritma tartışılması:

- 1- hangi işlemler gerçekleşince verilerin gelmesi gerektiği tartışıldı.

##### Alınan kararlar:

- 1- ilk olarak login işlemi yapılması gerektiği eğer bu işlem doğru girilmez ise verilere erişilememesi gerektiği kararına varıldı. Böylelikle güvenlik için bir önlem daha alınmış olundu.

#### VI. projenin içindeki uygulamaların entegrasyonu

- 1- projeye entegre edilecek mekanizmaların tartışılması.

##### Alınan kararlar:

- 1-veri tabanında verilerin api tarafıyla kullanılabilir bir yapı oluşturulması ve bu verilerin angular tarafına ulaşabilmesi için gerekli http methodları yazılması ve bu methodların anguların içerisindeki service ile bunların karşılanıp çekilebilir bir yapı oluşturulması ve bu yapıların TS kodları ile işlenebilir bir hale gelip html kodlarına entegrasyonu sağlanır.

#### Vii. oluşabilecek sorunlar

- 1-ileride oluşabilecek sorunlar konusunda tartışma.

##### Alınan kararlar:

- 1-kısaca şöyledir:

- Yoğun veri trafiği altında performans sorunları yaşanabilme riski
- Kullanıcıların yanlış veri girişi yapabilmesi veya yanlış verileri görüntüleyebilmesi gibi kullanıcı hataları riski
- Sistem güvenliğine yönelik potansiyel açıkların olabileme ihtimali,
- İnternet Erişimi sağlanamadığında sisteme erişimin olmaması

## 5.Analizler

### Gereksinim analizi

- Veri tabanı: Stok takibi için verilerin depolanması ve düzenlenmesi gerekmektedir. Bu nedenle, bir veri tabanı tasarımı gereklidir.
- Web API: Veri tabanından bilgi almak ve kullanıcı arayüzüne aktarmak için bir web API geliştirmeniz gerekebilir.
- Kullanıcı Arayüzü: Kullanıcılar sistemi kullanmak için bir web arayüzüne ihtiyaç duyarlar. Bu arayüz, grafikler, istatistikler ve otomatik öneriler içerebilir.
- Stok Yönetimi: Sistemin amacı stok takibi olduğundan, stok yönetimine ilişkin işlevselliğin sağlanması gerekmektedir.
- Ürün Yönetimi: Sistem, ürün bilgilerinin yönetilmesine de izin vermelidir. Bu, ürünlerin eklenmesi, düzenlenmesi ve silinmesini içerebilir.
- Admin Paneli: Kullanıcılar için bir admin paneli sağlamak, veri tabanını düzenlemelerine ve özelleştirmelerine olanak tanıyabilir.
- Filtreleme ve Erişim Kontrolü: Şirketlerin belirli kullanıcılara erişim izni vermesi gerekiyorsa, sistemin filtreleme ve erişim kontrolü özelliklerine sahip olması gerekebilir.

### Proje Temel Analizi

#### Proje Amacı:

Bu projenin amacı, stok kontrolü yapan firmaların ihtiyaçlarına yönelik bir web tabanlı stok kontrol sistemi geliştirmektir. Bu sistem sayesinde, firmaların stokları hakkında daha doğru ve detaylı bilgiye sahip olmaları, stok yönetimlerini daha etkili bir şekilde yapmaları, stoklarında meydana gelebilecek problemleri önceden tespit edebilmeleri ve dolayısıyla iş süreçlerini daha verimli hale getirebilmeleri hedeflenmektedir.

#### Proje Hedefi:

Bu projenin hedefi, stok kontrolü yapan firmaların ihtiyaçlarına yönelik bir web tabanlı stok kontrol sistemi geliřtirmek ve bu sistemi kullanıma sunarak müşteri memnuniyetini artırmak, iş süreçlerini optimize etmek, hata payını minimuma indirmek ve firmaların rekabet gücünü artırmaktır. Ayrıca, sistemde yer alan veri analizi ve raporlama özellikleri sayesinde, firmaların stoklarına ilişkin detaylı istatistiksel verileri elde etmeleri ve stok yönetimlerini daha verimli bir şekilde yapmaları hedeflenmektedir.

### **Proje Kapsamı:**

İřletme sahipleri, satın alma yöneticileri ve depo yöneticileri gibi stok yönetiminden sorumlu olan herkes bu sistemi kullanabilir. Bu sistem büyük ölçekli işletmeler için uygundur. İster tek bir depo isterse birden fazla depo ile çalışılsın, sistem tüm işletmelerin stok yönetim ihtiyaçlarını karşılamak için tasarlanmıştır. Ayrıca, bu sistem müşterilerin ihtiyaçlarına göre özelleřtirilebilir olduđu için, farklı sektörlerdeki işletmelerin de kullanabileceđi bir yapıdadır.

### **Swot Analizi**

#### **Güçlü Yönler (Strengths):**

- Basit arayüzü sayesinde kullanım kolaylığı
- İstatistiksel veriler ve grafiklerin sunulması sayesinde verilerin daha kolay takip edilmesi
- Esnek ve kişiselleřtirilebilir veri tabanı tasarımı sayesinde kullanıcıların kendi ihtiyaçlarına göre sistem özelleřtirmesi yapabilmesi
- Sistem içinde yetkilendirme ve erişim kontrolü sağlanması sayesinde kullanıcıların güvenliği artırılması
- Bilgilerin aktarılırken 'Token' sistemi ile şifrelenerek gönderilmesi

#### **Zayıf Yönler (Weaknesses):**

- Yoğun veri trafiđi altında performans sorunları yaşanabilme riski

- Kullanıcıların yanlış veri girişi yapabilmesi veya yanlış verileri görüntüleyebilmesi gibi kullanıcı hataları riski
- Sistem güvenliğine yönelik potansiyel açıkların olabilme ihtimali,
- İnternet Erişimi sağlanamadığında sisteme erişimin olmaması

#### **Fırsatlar (Opportunities):**

- Sistemin kullanıcılar tarafından kabul edilmesi durumunda pazar payının artması
- Sistemin yeni özelliklerle geliştirilerek daha fazla kullanıcının ihtiyacını karşılaması
- Sistemin performansı ve güvenliği için sürekli olarak geliştirmeler yapılması sayesinde rakiplerden farklılaşma fırsatı

#### **Tehditler (Threats):**

- Rakip stok kontrol sistemlerinin pazar payının artması
- Sistem güvenliği için oluşabilecek açıkların kullanıcıların güvenliğini riske atması
- Hızla değişen teknolojik trendlerin sistemin popülerliğini etkilemesi

#### **Sorun Analizi**

- Veri tabanı hataları: Veri tabanı hataları, sistemin çalışmasını engelleyebilir ve veri kaybına neden olabilir.
- Güvenlik açıkları: Güvenlik açıkları, sisteme yetkisiz erişim sağlanmasına neden olabilir ve hassas bilgilerin ele geçirilmesine yol açabilir.

- Performans sorunları: Sistem, yoğun veri trafiği altında performans sorunları yaşayabilir. Bu nedenle, yüksek trafiği yönetmek için sistem optimize edilmelidir.
- Kullanıcı hataları: Kullanıcılar yanlış veri girişi yapabilir veya yanlış verileri görüntüleyebilir. Bu nedenle, kullanıcı hatalarını önlemek için doğrulama kontrolleri gibi özellikler eklemek önemlidir.
- Veri kaybı: Veri kaybı, sistem çökmesi veya veri tabanı hataları gibi nedenlerden dolayı ortaya çıkabilir. Bu nedenle, veri yedekleme ve geri yükleme işlevleri eklemek önemlidir.

## **İhtiyaç Analizi**

- Güvenlik: Veri güvenliği, her türlü web tabanlı uygulamada kritik bir ihtiyaçtır. Bu nedenle, sisteminiz, veri kaybını önlemek ve hassas bilgileri korumak için güvenliği sağlamalıdır.
- Performans: Sistem, yüksek performans gerektiren bir uygulamadır. Kullanıcıların anlık verileri görüntülemesi gerekebilir, bu nedenle sistemin hızlı çalışması ve zamanında veri sağlaması önemlidir.
- Ölçeklenebilirlik: Sistemin, kullanıcı sayısı arttıkça ölçeklenebilir olması önemlidir. Birden fazla kullanıcı sistemde aynı anda çalışırken, performansı düşürmeden ve verileri güvenli bir şekilde işleyebilmelidir.
- Kullanılabilirlik: Kullanıcı arayüzünün kullanımı kolay ve sezgisel olmalıdır. Kullanıcıların stok takibi yaparken hızlı ve kolay bir şekilde istedikleri bilgilere erişmeleri gerekiyor. Stok yönetimi yaparken tasarımı responsive tasarlanmalıdır.



## 6.tasarım

PART II

## 7.kodlama

API kodlaması:

```
public class LoginResult
{
    public string Sk { get; set; }
    public string Ls { get; set; }
    public string Mk { get; set; }
    public string id { get; set; }
    public int id2 { get; set; }
    public List<object> Veri { get; set; }
    public List<object> sepet { get; set; }
    public List<object> siparis { get; set; }
    public int urunid { get; set; }
    public int adet { get; set; }
```

```

public int mk2 { get; set; }
    public int siparisid { get; set; }
    public int durumyeni { get; set; }
    public int dakika { get; set; }
    public int urun_id { get; set; }
    public int kategori_id { get; set; }
    public int marka_id { get; set; }
    public int urun_adet { get; set; }
    public int maliyet { get; set; }
    public int satis { get; set; }
    public string urun_ad { get; set; }
    public string aciklama { get; set; }
    public string kategori_ad { get; set; }
    public string marka_ad { get; set; }
    public string firstlast { get; set; }
    public string email1 { get; set; }
    public string email1eski { get; set; }
    public int ls1 { get; set; }
    public int mk1 { get; set; }
    public int id1 { get; set; }
    public string pass1 { get; set; }
    public int sirket_id { get; set; }

}

```

Açıklama:

Bu kısımda içerisine get ve set olaylarını yapabileceğim bir model tanımlamasıdır.

POST Alanı başlangıç noktası:

-----

```
[HttpPost("insertkategori")]
```

```
public IActionResult insertkategori([FromBody] LoginResult model)
```

```
{
```

```
    string queryinsert = @"insert into [AH_firmasi].[dbo].[kategoriler] values(@kategori_ad)";
```

```
    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
```

```
    using (SqlCommand command = new SqlCommand(queryinsert, con))
```

```

{
    command.Parameters.AddWithValue("@kategori_ad", model.kategori_ad);
    con.Open();
    command.ExecuteNonQuery();
}

return Ok();
}

[HttpPost("insertmarka")]
public IActionResult insertmarka([FromBody] LoginResult model)
{

    string queryinsert = @"insert into [AH_firmasi].[dbo].[markalar] values(@marka_ad)";

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
    using (SqlCommand command = new SqlCommand(queryinsert, con))
    {
        command.Parameters.AddWithValue("@marka_ad", model.marka_ad);
        con.Open();
        command.ExecuteNonQuery();
    }

    return Ok();
}

[HttpPost("yeniurun")]
public IActionResult yeniurun([FromBody] LoginResult model)
{

    string queryinsert = @"insert into [AH_firmasi].[dbo].[urunler] values
(@urun_id,@kategori_id,@marka_id,@urun_ad,@urun_adet,NULL,@maliyet,@satis,@aciklama)";

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))

```

```

using (SqlCommand command = new SqlCommand(queryinsert, con))
{
    command.Parameters.AddWithValue("@urun_id", model.urun_id);
    command.Parameters.AddWithValue("@kategori_id", model.kategori_id);
    command.Parameters.AddWithValue("@marka_id", model.marka_id);
    command.Parameters.AddWithValue("@urun_adet", model.urun_adet);
    command.Parameters.AddWithValue("@satis", model.satis);
    command.Parameters.AddWithValue("@malijet", model.malijet);
    command.Parameters.AddWithValue("@urun_ad", model.urun_ad);
    command.Parameters.AddWithValue("@aciklama", model.aciklama);

```

```

con.Open();
    command.ExecuteNonQuery();
}

```

```

return Ok();
}

```

```

[HttpPost("insert")]

```

```

public IActionResult Insert([FromBody] LoginResult model)

```

```

{

```

```

    DateTime datetimenow = DateTime.Now;

```

```

    string queryinsert = @"insert into siparisler
(urun_ID,durum_id,siparis_tarih,adet,anlik_fiyat,aciklama_id,magaza_id) values (@urunid, 4,
@datetimenow, @adet,0, 1,1)";

```

```

using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))

```

```

using (SqlCommand command = new SqlCommand(queryinsert, con))

```

```

{
    command.Parameters.AddWithValue("@urunid", model.urunid);
    command.Parameters.AddWithValue("@datetimenow", datetimenow);
    command.Parameters.AddWithValue("@adet", model.adet);

    con.Open();
    command.ExecuteNonQuery();
}

return Ok();
}

[HttpPost("yeniusers")]
public IActionResult yeniusers([FromBody] LoginResult model)
{
    string queryinsert = @"INSERT INTO [dbo].[Users]([firstLast],[email],[sk],[ls],[pass],[sirket_ID])
VALUES(@firstlast,@email,@mk,@ls,@pass,@sirket_id)";

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection")))
    using (SqlCommand command = new SqlCommand(queryinsert, con))
    {
        command.Parameters.AddWithValue("@firstlast", model.firstlast);
        command.Parameters.AddWithValue("@email", model.email1);
        command.Parameters.AddWithValue("@ls", model.ls1);
        command.Parameters.AddWithValue("@mk", model.mk1);
        command.Parameters.AddWithValue("@pass", model.pass1);
        command.Parameters.AddWithValue("@sirket_id", model.sirket_id);
    }
}

```

```
        con.Open();

        command.ExecuteNonQuery();

    }

    return Ok();
}
```

---

POST Alanı bitiş ve açıklaması: örnek olarak ilk kodu ele alınmıştır ve diğer örnekler de çok benzerdir.

POST isteğini karşılayan bir ASP.NET API Core Controller işlevini temsil ediyor. İsteğin `insertkategori` yoluna yapıldığı varsayılıyor. İsteğin gövdesinde (`FromBody`) bir `LoginResult` nesnesi alınıyor.

İşlevin amacı, `kategoriler` tablosuna yeni bir kategori eklemektir. Eklenen kategori adı `model` üzerinden alınır ve `@kategori\_ad` parametresi olarak kullanılır.

İşlevin çalışma mantığı şu şekildedir:

1. Ekleme sorgusunu içeren bir SQL sorgusu (`queryinsert`) tanımlanır.
2. `MyDBConnection2` adlı bağlantı dizesini kullanarak SQL Server veritabanına bağlantı oluşturulur (`SqlConnection`).
3. Bağlantı ve ekleme sorgusu (`SqlCommand`) için using blokları kullanılarak kaynakları yönetilir. Bu şekilde, bağlantı ve komut nesneleri işleri tamamlandığında otomatik olarak temizlenir.
4. Ekleme sorgusu için parametre tanımlanır: `@kategori\_ad` (eklenecek kategori adı).
5. Bağlantı açılır (`con.Open()`).
6. Ekleme sorgusu veritabanında çalıştırılır (`command.ExecuteNonQuery()`). Bu, `kategoriler` tablosuna yeni bir kayıt ekleyecektir.
7. İşlem başarıyla tamamlandığında, `Ok()` işlevi ile 200 (Başarılı) yanıt döndürülür.

PUT Alanı başlangıcı:

---

```

[HttpPut("updateuser")]
public IActionResult updateuser([FromBody] LoginResult model)
{

    string queryupdate = @" UPDATE[dbo].[Users] SET[firstLast] = @firstlast, [email] = @email,
[ls] = @ls, [pass] = @pass, [sirket_ID] = @sirket_id WHERE[email] = @email1eski;";

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection")))
    using (SqlCommand command = new SqlCommand(queryupdate, con))
    {

        command.Parameters.AddWithValue("@firstlast", model.firstlast);
        command.Parameters.AddWithValue("@email", model.email1);
        command.Parameters.AddWithValue("@ls", model.ls1);
        command.Parameters.AddWithValue("@email1eski", model.email1eski);
        command.Parameters.AddWithValue("@pass", model.pass1);
        command.Parameters.AddWithValue("@sirket_id", model.sirket_id);

        con.Open();
        command.ExecuteNonQuery();
    }

    return Ok();
}

[HttpPut("giris_tarih")]
public IActionResult giris_tarih([FromBody] LoginResult model)
{

    DateTime datetimenow = DateTime.Now;

    string queryupdate = @"UPDATE Users SET giris_tarihi = @durumyeni, adet=@adet
WHERE[email] = @email;";

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection")))

```

```

using (SqlCommand command = new SqlCommand(queryupdate, con))
{
    command.Parameters.AddWithValue("@durumyeni", model.durumyeni);
    command.Parameters.AddWithValue("@siparisid", model.siparisid);
    command.Parameters.AddWithValue("@adet", model.adet);

    con.Open();
    command.ExecuteNonQuery();
}

return Ok();
}

```

```

[HttpPut("update")]
public IActionResult update([FromBody] LoginResult model)
{
    DateTime datetimenow = DateTime.Now;

    string queryupdate = @"UPDATE siparisler SET durum_id = @durumyeni, adet=@adet
WHERE siparis_id=@siparisid;";

```

```

using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
using (SqlCommand command = new SqlCommand(queryupdate, con))
{
    command.Parameters.AddWithValue("@durumyeni", model.durumyeni);
    command.Parameters.AddWithValue("@siparisid", model.siparisid);
    command.Parameters.AddWithValue("@adet", model.adet);

    con.Open();
    command.ExecuteNonQuery();
}

```



```

        return Ok();
    }

    [HttpPut("updatesiparis")]
    public IActionResult updatesiparis([FromBody] LoginResult model)
    {

        DateTime datetimenow = DateTime.Now;

        string queryupdate = @"UPDATE siparisler SET adet=@adet WHERE siparis_id =@siparisid;";

        using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
        using (SqlCommand command = new SqlCommand(queryupdate, con))
        {

            command.Parameters.AddWithValue("@siparisid", model.siparisid);
            command.Parameters.AddWithValue("@adet", model.adet);

            con.Open();
            command.ExecuteNonQuery();
        }

        return Ok();
    }

    [HttpPut("updateurunler")]
    public IActionResult updateurunler([FromBody] LoginResult model)
    {

        DateTime datetimenow = DateTime.Now;

        string queryupdate = @"UPDATE [dbo].[urunler] SET [kategori_ID] = @kategori_id,[marka_ID]
= @marka_id,[urun_ad] = @urun_ad,

```

```
[maliyet] = @maliyet,[satis] = @satis,[aciklama] = @aciklama WHERE  
urun_ID=@urun_id";
```

```
using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))  
using (SqlCommand command = new SqlCommand(queryupdate, con))  
{  
  
    command.Parameters.AddWithValue("@urun_id", model.urun_id);  
    command.Parameters.AddWithValue("@kategori_id", model.kategori_id);  
    command.Parameters.AddWithValue("@marka_id", model.marka_id);  
    command.Parameters.AddWithValue("@satis", model.satis);  
    command.Parameters.AddWithValue("@maliyet", model.maliyet);  
    command.Parameters.AddWithValue("@urun_ad", model.urun_ad);  
    command.Parameters.AddWithValue("@aciklama", model.aciklama);  
  
    con.Open();  
    command.ExecuteNonQuery();  
}  
  
return Ok();  
}  
[HttpPut("adetekleurunler")]  
public IActionResult adetekleurunler([FromBody] LoginResult model)  
{
```

```
    DateTime datetimenow = DateTime.Now;  
  
    string queryupdate = @"UPDATE [dbo].[urunler] SET [urun_adet]=@urun_adet WHERE  
urun_ID=@urun_id";
```

```
using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))  
using (SqlCommand command = new SqlCommand(queryupdate, con))
```

```

{

    command.Parameters.AddWithValue("@urun_id", model.urun_id);
    command.Parameters.AddWithValue("@urun_adet", model.urun_adet);


    con.Open();
    command.ExecuteNonQuery();
}

return Ok();
}

[HttpPut("urunsil")]
public IActionResult urunsil([FromBody] LoginResult model)
{

    DateTime datetimenow = DateTime.Now;

    string queryupdate = @"UPDATE [dbo].[urunler] SET [kategori_ID] = -1 WHERE
urun_ID=@urun_id";

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
    using (SqlCommand command = new SqlCommand(queryupdate, con))
    {

        command.Parameters.AddWithValue("@urun_id", model.urun_id);


        con.Open();
        command.ExecuteNonQuery();
    }
}

```

```

    }

    return Ok();
}

[HttpPut("dakikaekle")]
public IActionResult dakikaekle([FromBody] LoginResult model)
{
    DateTime datetimenow = DateTime.Now;
    string queryupdate = @"UPDATE [LoginDB].[dbo].[Users]
        SET [dakika_1gun] = [dakika_1gun] + @dakika,
            [dakika_7gun] = [dakika_7gun] + @dakika
        WHERE [id] = @id";

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection")))
    using (SqlCommand command = new SqlCommand(queryupdate, con))
    {
        command.Parameters.AddWithValue("@id", model.id2);
        command.Parameters.AddWithValue("@dakika", model.dakika);

        con.Open();
        command.ExecuteNonQuery();
    }

    return Ok();
}

```

```

[HttpPost("getuser")]
public IActionResult getuser([FromBody] LoginResult model)
{
    string query = @"SELECT TOP (1000) * FROM [LoginDB].[dbo].[Users]where
sirket_ID=@sirket_id";

    List<object> getuser = new List<object>();

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection")))
    using (var cmd = new SqlCommand(query, con))
    {

        cmd.CommandType = CommandType.Text;

        cmd.Parameters.AddWithValue("@sirket_id", model.sirket_id); // model nesnesine doğru
        şekilde erişim sağladığınızdan emin olun

        con.Open();
        var reader = cmd.ExecuteReader();
        if (reader != null)
        {
            while (reader.Read())
            {
                var row = new Dictionary<string, object>();
                for (var i = 0; i < reader.FieldCount; i++)
                {
                    row[reader.GetName(i)] = reader.GetValue(i);
                }
                getuser.Add(row);
            }
        }
    }

    return Ok(getuser);
}

```

}

---

Put alanı bitiş ve açıklama:

Bu alanda yazılanlar user güncelleme, ürünün durum değişikliği tarihi,sepet hareketleri güncelleme, sipariş hareketleri, ürünü güncelleme,ürün adet değişimi,urun silmek için kategorisini -1 yapma, çalışanların çalışma dakikasını otomatik düzenleme, örnek olarak bir güncellemeyi şu şekilde açıklayabilirim:

PUT HTTP isteğini karşılayan bir ASP.NET Core API Controller işlevini temsil ediyor. İsteğin `dakikaekle` yoluna yapıldığı varsayılıyor. İsteğin gövdesinde (`FromBody`) bir `LoginResult` nesnesi alınıyor.

İşlevin amacı, bir kullanıcının belirli bir süreye (dakika) göre iki farklı sütununu güncellemektir: `dakika\_1gun` ve `dakika\_7gun`. Bu güncelleme, `Users` tablosunda `id` sütununu temel alarak yapılır.

İşlevin çalışma mantığı şu şekildedir:

1. İsteğin geldiği anın tarih ve saatini (`DateTime.Now`) alıyoruz.
2. Güncelleme sorgusunu içeren bir SQL sorgusu (`queryupdate`) tanımlıyoruz.
3. `MyDBConnection` adlı bağlantı dizesini kullanarak SQL Server veritabanına bağlantı oluşturuyoruz (`SqlConnection`).
4. Bağlantı ve güncelleme sorgusu (`SqlCommand`) için using blokları kullanarak kaynakları yönetiyoruz. Bu şekilde, bağlantı ve komut nesneleri işleri tamamlandığında otomatik olarak temizlenir.
5. Güncelleme sorgusu için parametreler tanımlıyoruz: `@id` (kullanıcı kimliği) ve `@dakika` (eklenecek dakika değeri).
6. Bağlantıyı açıyoruz (`con.Open()`).
7. Güncelleme sorgusunu veritabanında çalıştırıyoruz (`command.ExecuteNonQuery()`). Bu, kullanıcının `dakika\_1gun` ve `dakika\_7gun` sütunlarını istenen değerlerle güncelleyecektir.
8. İşlem başarıyla tamamlandığında, `Ok()` işlevi ile 200 (Başarılı) yanıt döndürülür.

---

Get alanı başlangıç:

```
[HttpGet("getlisance")]  
  
public IActionResult getlisance()
```

```

{
    string query = @"SELECT TOP(10)*FROM[LoginDB].[dbo].[lisanslar]";
    List<object> getlisance = new List<object>();

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
    using (var cmd = new SqlCommand(query, con))
    {
        cmd.CommandType = CommandType.Text;
        con.Open();
        var reader = cmd.ExecuteReader();
        if (reader != null)
        {
            while (reader.Read())
            {
                var row = new Dictionary<string, object>();
                for (var i = 0; i < reader.FieldCount; i++)
                {
                    row[reader.GetName(i)] = reader.GetValue(i);
                }
                getlisance.Add(row);
            }
        }
    }

    return Ok(getlisance);
}

[HttpGet("getdakika")]
public IActionResult getdakika()
{
    string query = @"SELECT *FROM users";
    List<object> getdakika = new List<object>();

```

```

using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection")))
using (var cmd = new SqlCommand(query, con))
{
    cmd.CommandType = CommandType.Text;
    con.Open();
    var reader = cmd.ExecuteReader();
    if (reader != null)
    {
        while (reader.Read())
        {
            var row = new Dictionary<string, object>();
            for (var i = 0; i < reader.FieldCount; i++)
            {
                row[reader.GetName(i)] = reader.GetValue(i);
            }
            getdakika.Add(row);
        }
    }
}

return Ok(getdakika);
}

[HttpGet("islemcisepet")]
public IActionResult islemcisepet()
{
    string query = @"SELECT * FROM [AH_firmasi].[dbo].[istekler] WHERE durum_id=1";
    List<object> islemcisepetveri = new List<object>();

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
    using (var cmd = new SqlCommand(query, con))

```



```

{
    cmd.CommandType = CommandType.Text;
    con.Open();
    var reader = cmd.ExecuteReader();
    if (reader != null)
    {
        while (reader.Read())
        {
            var row = new Dictionary<string, object>();
            for (var i = 0; i < reader.FieldCount; i++)
            {
                row[reader.GetName(i)] = reader.GetValue(i);
            }
            islemcisepetveri.Add(row);
        }
    }
}

return Ok(islemcisepetveri);
}

```

```

[HttpGet("sepet")]
public IActionResult sepet()
{
    string query = @"SELECT * FROM [AH_firnasi].[dbo].[istekler] WHERE sirket_ID=1 AND
durum_id=4";

    List<object> sepet = new List<object>();

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
    using (var cmd = new SqlCommand(query, con))
    {

```

```

        cmd.CommandType = CommandType.Text;
        con.Open();
        var reader = cmd.ExecuteReader();
        if (reader != null)
        {
            while (reader.Read())
            {
                var row = new Dictionary<string, object>();
                for (var i = 0; i < reader.FieldCount; i++)
                {
                    row[reader.GetName(i)] = reader.GetValue(i);
                }
                sepet.Add(row);
            }
        }
    }

    return Ok(sepet);
}

[HttpGet("gecmis")]
public IActionResult gecmis()
{
    string query = @"SELECT * FROM [AH_firnasi].[dbo].[gecmis_view] order by teslim_tarihi
desc";
    List<object> gecmis = new List<object>();

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
    using (var cmd = new SqlCommand(query, con))
    {
        cmd.CommandType = CommandType.Text;
        con.Open();
    }
}

```

```

        var reader = cmd.ExecuteReader();
        if (reader != null)
        {
            while (reader.Read())
            {
                var row = new Dictionary<string, object>();
                for (var i = 0; i < reader.FieldCount; i++)
                {
                    row[reader.GetName(i)] = reader.GetValue(i);
                }
                gecmis.Add(row);
            }
        }

        return Ok(gecmis);
    }

    [HttpGet("siparislerim")]
    public IActionResult siparislerim()
    {
        string query = @"SELECT * FROM [AH_firmasi].[dbo].[istekler] WHERE sirket_ID=1 and durum_id<4";

        List<object> siparisler = new List<object>();

        using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
        using (var cmd = new SqlCommand(query, con))
        {
            cmd.CommandType = CommandType.Text;
            con.Open();
            var reader = cmd.ExecuteReader();
            if (reader != null)

```

```

    {
        while (reader.Read())
        {
            var row = new Dictionary<string, object>();
            for (var i = 0; i < reader.FieldCount; i++)
            {
                row[reader.GetName(i)] = reader.GetValue(i);
            }
            siparisler.Add(row);
        }
    }

    return Ok(siparisler);
}

[HttpGet("tavsiye")]
public IActionResult tavsiye()
{
    string query = @"SELECT * FROM [AH_firmasi].[dbo].tavsiye ";
    List<object> tavsiye = new List<object>();

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
    using (var cmd = new SqlCommand(query, con))
    {
        cmd.CommandType = CommandType.Text;
        con.Open();
        var reader = cmd.ExecuteReader();
        if (reader != null)
        {
            while (reader.Read())
            {

```

```

        var row = new Dictionary<string, object>();
        for (var i = 0; i < reader.FieldCount; i++)
        {
            row[reader.GetName(i)] = reader.GetValue(i);
        }
        tavsiye.Add(row);
    }
}

return Ok(tavsiye);
}

[HttpGet("satisgunlerbirlesik")]
public IActionResult satisgunlerbirlesik()
{
    string query = @"SELECT urunler.[urun_ID], [tarih], SUM([adet]) AS toplam_adet,
[magaza_id],urun_ad,kategori_ID,marka_ID,urunler.maliyet,urunler.satis
FROM [AH_firmasi].[dbo].[satis] AS satis
INNER JOIN dbo.urunler ON satis.urun_ID = urunler.urun_ID
where kategori_ID!=-1
GROUP BY urunler.[urun_ID], [tarih],
[magaza_id],urun_ad,kategori_ID,marka_ID,urunler.maliyet,urunler.satis
order by toplam_adet desc
";

    List<object> satisgunlerbirlesik = new List<object>();

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
    using (var cmd = new SqlCommand(query, con))
    {
        cmd.CommandType = CommandType.Text;
        con.Open();
        var reader = cmd.ExecuteReader();
    }
}

```

```

        if (reader != null)
        {
            while (reader.Read())
            {
                var row = new Dictionary<string, object>();
                for (var i = 0; i < reader.FieldCount; i++)
                {
                    row[reader.GetName(i)] = reader.GetValue(i);
                }
                satisgunlerbirlesik.Add(row);
            }
        }
    }

    return Ok(satisgunlerbirlesik);
}

[HttpGet("sepetgecmis")]
public IActionResult sepetgecmis()
{
    string query = @"SELECT dbo.sepet.urun_ID, dbo.sepet.siparis_tarih, SUM([adet])as adet,
dbo.urunler.kategori_ID, dbo.urunler.marka_ID, dbo.urunler.urun_ad

FROM dbo.sepet INNER JOIN

dbo.urunler ON dbo.sepet.urun_ID = dbo.urunler.urun_ID

where kategori_ID!=-1

GROUP BY
sepet.urun_ID,sepet.siparis_tarih,urunler.kategori_ID,urunler.marka_ID,urunler.urun_ad

order by adet desc

";

    List<object> sepetgecmis = new List<object>();

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
    using (var cmd = new SqlCommand(query, con))

```

```

{
    cmd.CommandType = CommandType.Text;
    con.Open();
    var reader = cmd.ExecuteReader();
    if (reader != null)
    {
        while (reader.Read())
        {
            var row = new Dictionary<string, object>();
            for (var i = 0; i < reader.FieldCount; i++)
            {
                row[reader.GetName(i)] = reader.GetValue(i);
            }
            sepetgecmis.Add(row);
        }
    }
}

return Ok(sepetgecmis);
}

```

```

[HttpGet("hazirlanan")]
public IActionResult hazirlanan()
{
    string query = @"SELECT * FROM [AH_firnasi].[dbo].[istekler] WHERE sirket_ID=1 and durum_id=2 or durum_id=2";
    List<object> hazirlanan = new List<object>();

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
    using (var cmd = new SqlCommand(query, con))
    {

```

```

cmd.CommandType = CommandType.Text;
con.Open();
var reader = cmd.ExecuteReader();
if (reader != null)
{
    while (reader.Read())
    {
        var row = new Dictionary<string, object>();
        for (var i = 0; i < reader.FieldCount; i++)
        {
            row[reader.GetName(i)] = reader.GetValue(i);
        }
        hazirlanan.Add(row);
    }
}

return Ok(hazirlanan);
}

```

```

[HttpGet("veri")]
public IActionResult GetVeri()
{
    string query = @"SELECT * FROM [AH_firmasi].[dbo].[AH_firma_urunler]";
    List<object> veri = new List<object>();

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
    using (var cmd = new SqlCommand(query, con))
    {
        cmd.CommandType = CommandType.Text;
        con.Open();

```



```

var reader = cmd.ExecuteReader();
if (reader != null)
{
    while (reader.Read())
    {
        var row = new Dictionary<string, object>();
        for (var i = 0; i < reader.FieldCount; i++)
        {
            row[reader.GetName(i)] = reader.GetValue(i);
        }
        veri.Add(row);
    }
}

return Ok(veri);
}

```

```

[HttpGet("kategori")]
public IActionResult kategori()
{
    string query = @"SELECT * FROM [AH_firmasi].[dbo].[kategoriler]";
    List<object> kategori = new List<object>();

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
    using (var cmd = new SqlCommand(query, con))
    {
        cmd.CommandType = CommandType.Text;

```

```

        con.Open();

        var reader = cmd.ExecuteReader();

        if (reader != null)
        {
            while (reader.Read())
            {
                var row = new Dictionary<string, object>();

                for (var i = 0; i < reader.FieldCount; i++)
                {
                    row[reader.GetName(i)] = reader.GetValue(i);
                }

                kategori.Add(row);
            }
        }
    }

    return Ok(kategori);
}

```

```

[HttpGet("marka")]
public IActionResult marka()
{
    string query = @"SELECT * FROM [AH_firmasi].[dbo].[markalar]";

    List<object> marka = new List<object>();

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
    using (var cmd = new SqlCommand(query, con))
    {
        cmd.CommandType = CommandType.Text;

        con.Open();

        var reader = cmd.ExecuteReader();
    }
}

```

```

        if (reader != null)
        {
            while (reader.Read())
            {
                var row = new Dictionary<string, object>();
                for (var i = 0; i < reader.FieldCount; i++)
                {
                    row[reader.GetName(i)] = reader.GetValue(i);
                }
                marka.Add(row);
            }
        }
    }

    return Ok(marka);
}

[HttpGet("alis")]
public IActionResult alis()
{
    string query = @"SELECT * FROM [AH_firmasi].[dbo].[alis]";
    List<object> marka = new List<object>();

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
    using (var cmd = new SqlCommand(query, con))
    {
        cmd.CommandType = CommandType.Text;
        con.Open();
        var reader = cmd.ExecuteReader();
        if (reader != null)
        {
            while (reader.Read())

```

```

        {
            var row = new Dictionary<string, object>();
            for (var i = 0; i < reader.FieldCount; i++)
            {
                row[reader.GetName(i)] = reader.GetValue(i);
            }
            marka.Add(row);
        }
    }
}

return Ok(marka);
}

[HttpPut("deleteuser")]
public IActionResult deleteuser([FromBody] LoginResult model)
{
    DateTime datetimenow = DateTime.Now;
    string queryupdate = @"DELETE FROM [dbo].[Users] WHERE[id] =@id;";

    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection")))
    using (SqlCommand command = new SqlCommand(queryupdate, con))
    {
        command.Parameters.AddWithValue("@id", model.id1);

        con.Open();
        command.ExecuteNonQuery();
    }
}

```

```
        return Ok();  
    }  
}
```

Get alanı bitiş ve açıklama:

GET isteğini karşılayan bir ASP.NET Core API Controller işlevini temsil ediyor. İsteğin `getlisance` yoluna yapıldığı varsayılıyor.

İşlevin çalışma mantığı şu şekildedir:

1. Sorguyu içeren bir SQL sorgusu (`query`) tanımlanır. Bu sorgu, `LoginDB` veritabanındaki `lisanslar` tablosundan en fazla 10 kaydı seçer.
2. Bir `List<object>` olan `getlisance` oluşturulur. Bu liste, sorgudan alınan verileri tutmak için kullanılacaktır.
3. `MyDBConnection2` adlı bağlantı dizesini kullanarak SQL Server veritabanına bağlantı oluşturulur (`SqlConnection`).
4. Bağlantı ve komut nesneleri için using blokları kullanılarak kaynak yönetimi yapılır.
5. Komut nesnesi (`SqlCommand`) oluşturulur ve sorgu ve bağlantı bilgileri atanır.
6. Komut tipi `CommandType.Text` olarak ayarlanır.
7. Bağlantı açılır (`con.Open()`).
8. Sorgunun sonucunu okumak için `cmd.ExecuteReader()` kullanılır. Verileri okuyan bir veri okuyucu (`reader`) oluşturulur.
9. Veri okuyucu `null` değilse, her bir satırı okurken bir sözlük (`Dictionary<string, object>`) oluşturulur.
10. Her bir sütunu sözlüğe eklemek için bir döngü kullanılır. `reader.GetName(i)` ile sütunun adı alınır ve `reader.GetValue(i)` ile sütunun değeri alınır.
11. Oluşturulan sözlük, `getlisance` listesine eklenir.
12. Veri okuma tamamlandıktan sonra, bağlantı ve veri okuyucu otomatik olarak kapatılır.
13. `getlisance` listesi, `Ok()` işleviyle birlikte döndürülür. Bu, 200 (Başarılı) yanıtını içerir ve `getlisance` listesindeki verileri içerir.

Delete alanı başlangıç:

```
[HttpDelete("deletesiparis")]  
public IActionResult deletesiparis([FromBody] LoginResult model)  
{  
  
    DateTime datetimenow = DateTime.Now;  
  
    string queryupdate = @"UPDATE siparisler SET adet=@adet WHERE siparis_id=@siparisid;";
```

```
using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
using (SqlCommand command = new SqlCommand(queryupdate, con))
{

    command.Parameters.AddWithValue("@siparisid", model.siparisid);
    command.Parameters.AddWithValue("@adet", model.adet);


    con.Open();
    command.ExecuteNonQuery();
}

return Ok();
}
```

Delete alanı bitiş ve açıklama:

Bu kod, bir DELETE isteğini karşılayan bir ASP.NET Core API Controller işlevini temsil ediyor. İsteğin `deletesiparis` yoluna yapıldığı varsayılıyor. İsteğin gövdesinde (`FromBody`) bir `LoginResult` nesnesi alınıyor.

İşlevin amacı, `siparisler` tablosunda belirli bir siparişin adet değerini güncellemektir. Güncelleme işlemi `siparis\_id` değerine göre yapılır.

İşlevin çalışma mantığı şu şekildedir:

1. Güncelleme sorgusunu içeren bir SQL sorgusu (`queryupdate`) tanımlanır. Bu sorgu, `siparisler` tablosunda belirli bir siparişin adet değerini günceller.
2. Bir `SqlConnection` nesnesi oluşturulur ve `MyDBConnection2` adlı bağlantı dizesini kullanarak SQL Server veritabanına bağlantı oluşturulur.
3. Bağlantı ve komut nesneleri için using blokları kullanılarak kaynak yönetimi yapılır.
4. Komut nesnesi (`SqlCommand`) oluşturulur ve sorgu ve bağlantı bilgileri atanır.
5. Parametreler (`@siparisid` ve `@adet`) `model` üzerinden alınır ve ilgili parametrelerle eşleştirilir.
6. Bağlantı açılır (`con.Open()`).
7. Güncelleme sorgusu veritabanında çalıştırılır (`command.ExecuteNonQuery()`). Bu, `siparisler` tablosundaki belirli siparişin adet değerini güncelleyecektir.
8. İşlem başarıyla tamamlandığında, `Ok()` işlevi ile 200 (Başarılı) yanıt döndürülür.

Login alanı başlangıç:

```
[EnableCors("AllowOrigin")]
```

```
[HttpPost("LoginUser")]
```

```
public IActionResult Login(Login user)
```

```
{
```

```
    string query = @"SELECT TOP (1) [sk] FROM [LoginDB].[dbo].[Users] WHERE email=@Email";
```

```
    string sk = "";
```

```
    string query3 = @"SELECT TOP (1) [sirket_ID] FROM [LoginDB].[dbo].[Users] WHERE  
email=@Email";
```

```
    string mk = "";
```

```
    string query6 = @"SELECT TOP (1) [id] FROM [LoginDB].[dbo].[Users] WHERE email=@Email";
```

```
    string id = "";
```

```
    string query1 = @"SELECT * FROM [AH_firmasi].[dbo].[AH_firma_urunler]";
```

```
    List<object> veri = new List<object>();
```

```
    string query2 = @"SELECT TOP (1) [Is] FROM [LoginDB].[dbo].[Users] WHERE email=@Email";
```

```
    string Is = "";
```

```
    string query4 = @"SELECT *FROM [AH_firmasi].[dbo].[siparisler] where magaza_id=@mk and  
durum_id=4";//96 satırdaki mk bilgisine göre
```

```
    List<object> sepet = new List<object>();
```

```
    string query5 = @"SELECT *FROM [AH_firmasi].[dbo].[siparisler] where magaza_id=@mk and  
durum_id=1";//96 satırdaki mk bilgisine göre
```

```
    List<object> siparis = new List<object>();
```

```

using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
using (var cmd = new SqlCommand(query1, con))
{
    cmd.CommandType = CommandType.Text;
    con.Open();
    var reader = cmd.ExecuteReader();
    if (reader != null)
    {
        while (reader.Read())
        {
            var row = new Dictionary<string, object>();
            for (var i = 0; i < reader.FieldCount; i++)
            {
                row[reader.GetName(i)] = reader.GetValue(i);
            }
            veri.Add(row);
        }
    }
}

```

```

using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection")))
{
    con.Open();

    sk = new SqlCommand(query, con)
    {
        CommandType = CommandType.Text,
        Parameters = { new SqlParameter("@Email", user.email) }
    }.ExecuteScalar()?.ToString();
}

```



```

mk = new SqlCommand(query3, con)
{
    CommandType = CommandType.Text,
    Parameters = { new SqlParameter("@Email", user.email) }
}.ExecuteScalar()?.ToString();

id = new SqlCommand(query6, con)
{
    CommandType = CommandType.Text,
    Parameters = { new SqlParameter("@Email", user.email) }
}.ExecuteScalar()?.ToString();

ls = new SqlCommand(query2, con)
{
    CommandType = CommandType.Text,
    Parameters = { new SqlParameter("@Email", user.email) }
}.ExecuteScalar()?.ToString();
}

if (mk != null)
{
    using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
    using (var cmd = new SqlCommand(query4, con))
    {
        cmd.CommandType = CommandType.Text;
        cmd.Parameters.AddWithValue("@Mk", mk); // move this line up
        con.Open();
        var reader = cmd.ExecuteReader();
        if (reader != null)
        {
            while (reader.Read())

```

```

    {
        var row2 = new Dictionary<string, object>();
        for (var i = 0; i < reader.FieldCount; i++)
        {
            row2[reader.GetName(i)] = reader.GetValue(i);
        }
        sepet.Add(row2);
    }
}

using (var con = new SqlConnection(_config.GetConnectionString("MyDBConnection2")))
using (var cmd = new SqlCommand(query5, con))
{
    cmd.CommandType = CommandType.Text;
    cmd.Parameters.AddWithValue("@Mk", mk); // move this line up
    con.Open();
    var reader = cmd.ExecuteReader();
    if (reader != null)
    {
        while (reader.Read())
        {
            var row2 = new Dictionary<string, object>();
            for (var i = 0; i < reader.FieldCount; i++)
            {
                row2[reader.GetName(i)] = reader.GetValue(i);
            }
            siparis.Add(row2);
        }
    }
}
}
}

```

```

else
{
    return Ok("Failure");
}

var userAvailable = _context.Users.FirstOrDefault(u => u.email == user.email && u.pass ==
user.pass);

if (userAvailable != null)
{
    var loginResult = new LoginResult { Sk = sk, Ls = ls, Veri=veri, Mk=mk ,id=id,
sepet=sepet,siparis=siparis}; // set the ls value here

    return Ok(loginResult);
}
else
{
    return Ok("Failure");
}}

```

Login bitiş ve açıklama:

Bu kod bir ASP.NET Core API Controller sınıfını temsil ediyor. `LoginUser` adında bir HTTP POST isteğiyle erişilebilen bir işlevi içeriyor. Ayrıca `EnableCors` özniteliği, CORS (Cross-Origin Resource Sharing) politikalarını uygulamak için kullanılıyor.

İşlevin amacı kullanıcının giriş yapmasını sağlamaktır. İşlevin çalışma mantığı şu şekildedir:

1. İlk olarak, `Login` adında bir model parametresi alır. Bu model, kullanıcının e-posta ve parola bilgilerini içerir.
2. İlgili SQL sorguları tanımlanır. Bu sorgular, veritabanında kullanıcının e-postasına göre ilgili bilgileri çekmek için kullanılır.
3. Bağlantı dizesi (`MyDBConnection2`) kullanılarak SQL Server veritabanına bağlantı oluşturulur.
4. İlk SQL sorgusu (`query1`), `AH\_firnasi.dbo.AH\_firma\_urunler` tablosundaki verileri çekmek için kullanılır. Elde edilen veriler, `veri` adlı bir liste nesnesine eklenir.
5. İkinci SQL sorgusu (`query`), `LoginDB.dbo.Users` tablosundan kullanıcının `email` değerine göre `sk` (security key) bilgisini çeker.
6. Üçüncü SQL sorgusu (`query3`), kullanıcının `email` değerine göre `sirket\_ID` bilgisini çeker.

7. Dördüncü SQL sorgusu (`query6`), kullanıcının `email` değerine göre `id` bilgisini çeker.
8. Beşinci SQL sorgusu (`query2`), kullanıcının `email` değerine göre `ls` (login status) bilgisini çeker.
9. Bağlantı açılır ve sorgular çalıştırılır. İlgili değerler ilgili değişkenlere atanır.
10. Eğer `mk` değeri null değilse (kullanıcı mevcutsa), `query4` ve `query5` ile ilgili sipariş ve sepet bilgileri çekilir. Bu bilgiler, `sepet` ve `siparis` adlı liste nesnelere eklenir.
11. `\_context.Users` üzerinde kullanıcının e-posta ve parolasına göre kullanıcının veritabanında mevcut olup olmadığı kontrol edilir. Eğer mevcutsa, `LoginResult` nesnesi oluşturulur ve ilgili değerler atanır.
12. Başarılı bir giriş durumunda, `Ok()` işlevi ile 200 (Başarılı) yanıt döndürülür ve `loginResult` nesnesi ile kullanıcı bilgileri döndürülür.
13. Başarısız bir giriş durumunda, `Ok()` işlevi ile 200 (Başarılı) yanıt döndürülür ve "Failure" metni döndürülür.

Bu kod,

kullanıcının giriş yapmasını sağlamak için veritabanı sorgularıyla çalışır. Kullanıcının doğru e-posta ve parola bilgilerini sağlaması durumunda, ilgili kullanıcı bilgilerini döndürür.

Appsetting.json :

```
{
  "ConnectionStrings": {
    "MyDBConnection": "Data Source=ITOCH\\ITOCH;Initial Catalog=LoginDB;Integrated Security=True",
    "MyDBConnection2": "Data Source=ITOCH\\ITOCH;Initial Catalog=AH_firmasi;Integrated Security=True"
  },
  "Logging": {
    "LogLevel": {
      "Default": "Information",
      "Microsoft": "Warning",
      "Microsoft.Hosting.Lifetime": "Information"
    }
  },
  "AllowedHosts": "*"
}
```

Açıklama:

JSON dosyasıdır ve bir ASP.NET Core uygulamasının yapılandırma bilgilerini içerir. Bu dosyada aşağıdaki bilgiler bulunmaktadır:

- `ConnectionStrings`: Veritabanı bağlantı dizelerini içeren bölüm. `MyDBConnection` ve `MyDBConnection2` adında iki ayrı bağlantı dizesi bulunmaktadır. Bu bağlantı dizeleri, SQL Server veritabanlarına bağlanmak için kullanılır. İlgili veritabanlarının bilgileri, `Data Source` ve `Initial Catalog` parametrelerinde belirtilir. Bağlantıların Windows kimlik doğrulamasıyla yapılacağı belirtilmiştir (`Integrated Security=True`).
- `Logging`: Günlükleme (logging) yapılandırması için kullanılan bölüm. Burada varsayılan günlükleme seviyesi (`Default`) ve Microsoft bileşenleri için günlükleme seviyeleri (`Microsoft` ve `Microsoft.Hosting.Lifetime`) belirtilmiştir.
- `AllowedHosts`: Uygulamanın hangi istemcilerden gelen isteklere izin vereceğini belirten bölüm. `"\*"` değeri, tüm istemcilere izin verildiği anlamına gelir. Bu, CORS (Cross-Origin Resource Sharing) politikalarını uygulamaya izin verir.

Bu yapılandırma dosyası, uygulamanın veritabanı bağlantılarını, günlükleme seviyelerini ve izin verilen istemcileri belirlemek için kullanılır.  
Sql kodlarımız:

AH\_firması database:

Tablo oluşturma kodları:

```
USE [AH_firmasi]
```

```
GO
```

```
/***** Object: Table [dbo].[aciklamalar]  Script Date: 26.05.2023 03:53:14 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[aciklamalar](
```

```
[aciklama_id] [int] NOT NULL,
```

```
[aciklama] [varchar](50) NULL,
```

```
CONSTRAINT [PK_aciklamalar] PRIMARY KEY CLUSTERED
```

```
(
```

```
[aciklama_id] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)  
ON [PRIMARY]
```

```
) ON [PRIMARY]
```

```
GO
```

```
USE [AH_firmasi]
```

```
GO
```

/\*\*\*\*\* Object: Table [dbo].[alis] Script Date: 26.05.2023 03:55:31 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[alis](

[urun\_ID] [int] NULL,

[tarih] [date] NULL,

[adet] [int] NULL

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[alis] WITH CHECK ADD CONSTRAINT [FK\_alis\_urunler] FOREIGN KEY([urun\_ID])

REFERENCES [dbo].[urunler] ([urun\_ID])

GO

ALTER TABLE [dbo].[alis] CHECK CONSTRAINT [FK\_alis\_urunler]

GO

USE [AH\_firmasi]

GO

/\*\*\*\*\* Object: Table [dbo].[durumlar] Script Date: 26.05.2023 03:57:25 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

```
CREATE TABLE [dbo].[durumlar](
[durum_id] [int] NOT NULL,
[durum] [varchar](50) NULL,
CONSTRAINT [PK_durumlar] PRIMARY KEY CLUSTERED
(
[durum_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
GO
USE [AH_firmasi]
GO
```

/\*\*\*\*\* Object: Table [dbo].[gecmis] Script Date: 26.05.2023 03:57:37 \*\*\*\*\*/

```
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TABLE [dbo].[gecmis](
[urun_id] [int] NULL,
[teslim_tarihi] [date] NULL,
[adet] [int] NULL,
[anlik_fiyat] [int] NULL,
[magaza_id] [int] NULL,
[siparis_tarih] [date] NULL,
[durum_id] [int] NULL,
[anlik_maliyet] [int] NULL
) ON [PRIMARY]
```

GO

```
ALTER TABLE [dbo].[gecmis] WITH CHECK ADD CONSTRAINT [FK_gecmis_urunler] FOREIGN  
KEY([urun_id])
```

```
REFERENCES [dbo].[urunler] ([urun_ID])
```

GO

```
ALTER TABLE [dbo].[gecmis] CHECK CONSTRAINT [FK_gecmis_urunler]
```

GO

```
USE [AH_firmasi]
```

GO

```
/***** Object: Table [dbo].[kategoriler] Script Date: 26.05.2023 03:57:48 *****/
```

```
SET ANSI_NULLS ON
```

GO

```
SET QUOTED_IDENTIFIER ON
```

GO

```
CREATE TABLE [dbo].[kategoriler](  
[kategori_ID] [int] IDENTITY(1,1) NOT NULL,  
[kategori_AD] [varchar](100) NULL,  
CONSTRAINT [PK_kategoriler] PRIMARY KEY CLUSTERED  
(  
[kategori_ID] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)  
ON [PRIMARY]  
) ON [PRIMARY]  
GO  
USE [AH_firmasi]
```



GO

/\*\*\*\*\* Object: Table [dbo].[markalar] Script Date: 26.05.2023 03:58:02 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

```
CREATE TABLE [dbo].[markalar](
    [marka_ID] [int] IDENTITY(1,1) NOT NULL,
    [marka_AD] [varchar](50) NULL,
    CONSTRAINT [PK_markalar] PRIMARY KEY CLUSTERED
(
    [marka_ID] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
GO
USE [AH_firmasi]
GO
```

/\*\*\*\*\* Object: Table [dbo].[satis] Script Date: 26.05.2023 03:58:13 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

```
CREATE TABLE [dbo].[satis](
    [urun_ID] [int] NULL,
```

```
[tarih] [date] NULL,  
[adet] [int] NULL,  
[magaza_id] [int] NULL,  
[siparis_tarih] [date] NULL  
) ON [PRIMARY]  
GO
```

```
ALTER TABLE [dbo].[satis] WITH CHECK ADD CONSTRAINT [FK_satis_urunler] FOREIGN  
KEY([urun_ID])  
REFERENCES [dbo].[urunler] ([urun_ID])  
GO
```

```
ALTER TABLE [dbo].[satis] CHECK CONSTRAINT [FK_satis_urunler]  
GO  
USE [AH_firmasi]  
GO
```

```
/***** Object: Table [dbo].[sepet] Script Date: 26.05.2023 03:58:22 *****/  
SET ANSI_NULLS ON  
GO
```

```
SET QUOTED_IDENTIFIER ON  
GO
```

```
CREATE TABLE [dbo].[sepet](  
[urun_ID] [int] NULL,  
[siparis_tarih] [date] NULL,  
[adet] [int] NULL  
) ON [PRIMARY]  
GO  
USE [AH_firmasi]
```

GO

/\*\*\*\*\* Object: Table [dbo].[siparisler] Script Date: 26.05.2023 03:58:31 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

```
CREATE TABLE [dbo].[siparisler](
[siparis_id] [int] IDENTITY(1,1) NOT NULL,
[urun_ID] [int] NULL,
[durum_id] [int] NULL,
[siparis_tarih] [date] NULL,
[adet] [int] NULL,
[anlik_fiyat] [int] NULL,
[aciklama_id] [int] NULL,
[magaza_id] [int] NULL,
[anlik_maliyet] [int] NULL,
CONSTRAINT [PK_siparisler] PRIMARY KEY CLUSTERED
(
[siparis_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)
ON [PRIMARY]
) ON [PRIMARY]
```

GO

```
ALTER TABLE [dbo].[siparisler] WITH CHECK ADD CONSTRAINT [FK_siparisler_aciklamalar] FOREIGN
KEY([aciklama_id])
```

```
REFERENCES [dbo].[aciklamalar] ([aciklama_id])
```

GO

```
ALTER TABLE [dbo].[siparisler] CHECK CONSTRAINT [FK_siparisler_aciklamalar]
```

```
GO
```

```
ALTER TABLE [dbo].[siparisler] WITH CHECK ADD CONSTRAINT [FK_siparisler_durumlar] FOREIGN  
KEY([durum_id])
```

```
REFERENCES [dbo].[durumlar] ([durum_id])
```

```
GO
```

```
ALTER TABLE [dbo].[siparisler] CHECK CONSTRAINT [FK_siparisler_durumlar]
```

```
GO
```

```
ALTER TABLE [dbo].[siparisler] WITH CHECK ADD CONSTRAINT [FK_siparisler_urunler] FOREIGN  
KEY([urun_ID])
```

```
REFERENCES [dbo].[urunler] ([urun_ID])
```

```
GO
```

```
ALTER TABLE [dbo].[siparisler] CHECK CONSTRAINT [FK_siparisler_urunler]
```

```
GO
```

```
USE [AH_firmasi]
```

```
GO
```

```
/***** Object: Table [dbo].[tavsiye] Script Date: 26.05.2023 03:58:44 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[tavsiye](
```

```
[Messages_Id] [int] IDENTITY(1,1) NOT NULL,
```

```
[degisken] [varchar](100) NOT NULL,  
[urun_tavsiyesi] [varchar](150) NULL,  
CONSTRAINT [PK_tavsiye] PRIMARY KEY CLUSTERED  
(  
[Messages_Id] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)  
ON [PRIMARY]  
) ON [PRIMARY]  
GO  
USE [AH_firmasi]  
GO
```

/\*\*\*\*\* Object: Table [dbo].[urunler] Script Date: 26.05.2023 03:58:59 \*\*\*\*\*/

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[urunler](  
[urun_ID] [int] NOT NULL,  
[kategori_ID] [int] NULL,  
[marka_ID] [int] NULL,  
[urun_ad] [varchar](50) NULL,  
[urun_adet] [int] NULL,  
[iss] [int] NULL,  
[maliyet] [float] NULL,  
[satis] [float] NULL,  
[aciklama] [varchar](100) NULL,  
CONSTRAINT [PK_urunler] PRIMARY KEY CLUSTERED  
(
```

[urun\_ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF,  
ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF)  
ON [PRIMARY]

) ON [PRIMARY]

GO

ALTER TABLE [dbo].[urunler] WITH CHECK ADD CONSTRAINT [FK\_urunler\_kategoriler] FOREIGN  
KEY([kategori\_ID])

REFERENCES [dbo].[kategoriler] ([kategori\_ID])

GO

ALTER TABLE [dbo].[urunler] CHECK CONSTRAINT [FK\_urunler\_kategoriler]

GO

ALTER TABLE [dbo].[urunler] WITH CHECK ADD CONSTRAINT [FK\_urunler\_markalar] FOREIGN  
KEY([marka\_ID])

REFERENCES [dbo].[markalar] ([marka\_ID])

GO

ALTER TABLE [dbo].[urunler] CHECK CONSTRAINT [FK\_urunler\_markalar]

GO

View kodları:

USE [AH\_firmasi]

GO

/\*\*\*\*\* Object: View [dbo].[AH\_firma\_urunler] Script Date: 26.05.2023 04:01:57 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE VIEW [dbo].[AH\_firma\_urunler]

AS

SELECT     dbo.urunler.urun\_ID, dbo.urunler.kategori\_ID, dbo.kategoriler.kategori\_AD,  
            dbo.urunler.marka\_ID, dbo.markalar.marka\_AD, dbo.urunler.urun\_ad, dbo.urunler.urun\_adet,  
            dbo.urunler.maliyet, dbo.urunler.satis,

            dbo.urunler.aciklama, dbo.urunler.iss

FROM       dbo.kategoriler INNER JOIN

            dbo.urunler ON dbo.kategoriler.kategori\_ID = dbo.urunler.kategori\_ID INNER JOIN

            dbo.markalar ON dbo.urunler.marka\_ID = dbo.markalar.marka\_ID

GO

USE [AH\_firmasi]

GO

/\*\*\*\*\* Object: View [dbo].[alınanlar]   Script Date: 26.05.2023 04:02:09 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE VIEW [dbo].[alınanlar]

AS

SELECT     dbo.alis.urun\_ID, dbo.urunler.urun\_turu, dbo.markalar.marka\_AD, dbo.urunler.urun\_ad,  
            dbo.alis.adet, dbo.alis.tarih, dbo.urunler.maliyet, dbo.urunler.satis

FROM       dbo.alis INNER JOIN

            dbo.urunler ON dbo.alis.urun\_ID = dbo.urunler.urun\_ID INNER JOIN

            dbo.markalar ON dbo.urunler.marka\_ID = dbo.markalar.marka\_ID

GO

GO

USE [AH\_firmasi]

GO

/\*\*\*\*\* Object: View [dbo].[gecmis\_view] Script Date: 26.05.2023 04:02:23 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE VIEW [dbo].[gecmis\_view]

AS

SELECT     dbo.gecmis.teslim\_tarihi, dbo.gecmis.adet, dbo.gecmis.siparis\_tarih,  
dbo.kategoriler.kategori\_AD, dbo.markalar.marka\_AD, dbo.urunler.urun\_ID, dbo.urunler.urun\_ad,  
dbo.urunler.aciklama, dbo.durumlar.durum,  
magazalar.dbo.magazalar.magaza\_ad, dbo.urunler.kategori\_ID, dbo.urunler.marka\_ID,  
dbo.gecmis.anlik\_fiyat AS satis, dbo.durumlar.durum\_id, dbo.gecmis.anlik\_maliyet AS maliyet

FROM       dbo.durumlar INNER JOIN

          dbo.gecmis ON dbo.durumlar.durum\_id = dbo.gecmis.durum\_id INNER JOIN

          dbo.urunler ON dbo.gecmis.urun\_id = dbo.urunler.urun\_ID INNER JOIN

          dbo.kategoriler ON dbo.urunler.kategori\_ID = dbo.kategoriler.kategori\_ID INNER JOIN

          dbo.markalar ON dbo.urunler.marka\_ID = dbo.markalar.marka\_ID INNER JOIN

          magazalar.dbo.magazalar ON magazalar.dbo.magazalar.magaza\_id =  
dbo.gecmis.magaza\_id

GO

GO

USE [AH\_firmasi]

GO

/\*\*\*\*\* Object: View [dbo].[istekler] Script Date: 26.05.2023 04:02:44 \*\*\*\*\*/

SET ANSI\_NULLS ON



GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE VIEW [dbo].[istekler]

AS

SELECT TOP (1000) i.urun\_ID, i.urun\_ad, i.marka\_AD, i.kategori\_AD, i.aciklama, i.siparis\_tarih,  
i.siparis\_adeti, i.urun\_adet, i.durum, s.Sirket\_AD, i.siparis\_id, i.magaza\_id, i.durum\_id, s.sirket\_ID,  
dbo.urunler.satis, dbo.urunler.iss

FROM dbo.istekler0 AS i INNER JOIN

LoginDB.dbo.sirketad AS s ON i.magaza\_id = s.sirket\_ID INNER JOIN

dbo.urunler ON i.urun\_ID = dbo.urunler.urun\_ID

GO

USE [AH\_firmasi]

GO

/\*\*\*\*\* Object: View [dbo].[istekler0] Script Date: 26.05.2023 04:02:55 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE VIEW [dbo].[istekler0]

AS

SELECT dbo.urunler.urun\_ID, dbo.urunler.urun\_ad, dbo.markalar.marka\_AD,  
dbo.kategoriler.kategori\_AD, dbo.aciklamalar.aciklama, dbo.siparisler.siparis\_tarih,  
dbo.siparisler.adet AS siparis\_adeti, dbo.urunler.urun\_adet,

dbo.durumlar.durum, dbo.siparisler.magaza\_id, dbo.siparisler.urun\_ID AS Expr1,  
dbo.siparisler.siparis\_id, dbo.durumlar.durum\_id

FROM dbo.aciklamalar INNER JOIN

dbo.siparisler ON dbo.aciklamalar.aciklama\_id = dbo.siparisler.aciklama\_id INNER JOIN

```
dbo.durumlar ON dbo.siparisler.durum_id = dbo.durumlar.durum_id INNER JOIN
dbo.urunler ON dbo.siparisler.urun_ID = dbo.urunler.urun_ID INNER JOIN
dbo.markalar ON dbo.urunler.marka_ID = dbo.markalar.marka_ID INNER JOIN
dbo.kategoriler ON dbo.urunler.kategori_ID = dbo.kategoriler.kategori_ID
```

GO

GO

USE [AH\_firmasi]

GO

/\*\*\*\*\* Object: View [dbo].[satilanlar] Script Date: 26.05.2023 04:03:07 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE VIEW [dbo].[satilanlar]

AS

```
SELECT    dbo.satis.urun_ID, dbo.urunler.urun_turu, dbo.markalar.marka_AD,
dbo.urunler.urun_ad, dbo.satis.adet, dbo.satis.tarih, dbo.urunler.maliyet, dbo.urunler.satis
```

```
FROM      dbo.satis INNER JOIN
```

```
          dbo.urunler ON dbo.satis.urun_ID = dbo.urunler.urun_ID INNER JOIN
```

```
          dbo.markalar ON dbo.urunler.marka_ID = dbo.markalar.marka_ID
```

GO

Trigger kodları:

USE [AH\_firmasi]

GO

/\*\*\*\*\* Object: Trigger [dbo].[tavsiye\_Trigger] Script Date: 26.05.2023 04:05:16 \*\*\*\*\*/

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TRIGGER [dbo].[tavsiye_Trigger]
```

```
ON [dbo].[satis]
```

```
AFTER INSERT
```

```
AS
```

```
BEGIN
```

```
    DECLARE @en_cok_satan_urun INT;
```

```
    DECLARE @en_az_satan_urun INT;
```

```
    DECLARE @en_cok_satan_kategori INT;
```

```
    DECLARE @en_az_satan_kategori INT;
```

```
    DECLARE @en_cok_satan_marka INT;
```

```
    DECLARE @en_az_satan_marka INT;
```

```
    DECLARE @bu_hafta_en_cok_satan_urun INT;
```

```
    DECLARE @bu_hafta_en_az_satan_urun INT;
```

```
    DECLARE @son_iki_hafta_satilmayan_urunler varchar(100);
```

```
declare @sayi int;
```

```
-- en çok satan ürün
```

```
SELECT TOP 1 @en_cok_satan_urun = urun_ID
```

```
FROM dbo.satis
```

```
GROUP BY urun_Id
```

```
ORDER BY SUM(adet) DESC;
```

```
--update
```

```
UPDATE tavsiye
```

```
SET degisken = (select urun_ad from urunler where urun_ID=@en_cok_satan_urun)
```

```
WHERE Messages_Id = 2;
```

```
-- en az satan ürün
```

```
SELECT TOP 1 @en_az_satan_urun = urun_Id
```

```
FROM dbo.satis
```

```
GROUP BY urun_Id
```

```
ORDER BY SUM(adet);
```

```
--update
```

```
UPDATE tavsiye
```

```
SET degisken = (select urun_ad from urunler where urun_ID=@en_az_satan_urun)
```

```
WHERE Messages_Id = 3;
```

```
-- en çok satan kategori
```

```
SELECT TOP 1 @en_cok_satan_kategori = kategori_ID
```

```
FROM dbo.urunler
```

```
INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id
```

```
GROUP BY kategori_ID
```

```
ORDER BY SUM(adet) DESC;
```

```
--update
```

```
UPDATE tavsiye
```

```
SET degisken = (select kategori_AD from kategoriler where kategori_ID=@en_cok_satan_kategori)
```

```
WHERE Messages_Id = 4;
```

```
-- en az satan kategori
```

```
SELECT TOP 1 @en_az_satan_kategori = kategori_ID
```

```
FROM dbo.urunler
```

```
INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id
```

```
GROUP BY kategori_ID
ORDER BY SUM(adet);

--update
UPDATE tavsiye
SET degisken = (select kategori_AD from kategoriler where kategori_ID=@en_az_satan_kategori)
WHERE Messages_Id = 5;
```

```
-- en çok satan marka
SELECT TOP 1 @en_cok_satan_marka = marka_ID
FROM dbo.urunler
INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id
GROUP BY marka_ID
ORDER BY SUM(adet) DESC;
```

```
--update
UPDATE tavsiye
SET degisken = (select marka_AD from markalar where marka_ID=@en_cok_satan_marka)
WHERE Messages_Id = 6;
```

```
-- en az satan marka
SELECT TOP 1 @en_az_satan_marka = marka_ID
FROM dbo.urunler
INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id
GROUP BY marka_ID
ORDER BY SUM(adet);
```

```
--update
UPDATE tavsiye
SET degisken = (select marka_AD from markalar where marka_ID=@en_az_satan_marka)
WHERE Messages_Id = 7;
```

```
-- bu hafta en çok satan ürün

SELECT TOP 1 @bu_hafta_en_cok_satan_urun = urun_Id
FROM dbo.satis
WHERE tarih >= DATEADD(week, -1, GETDATE())
GROUP BY urun_Id
ORDER BY (SUM(adet) / (SELECT SUM(adet) FROM dbo.satis WHERE tarih < DATEADD(week, -1, GETDATE()))) * 100) DESC;
```

--update

```
UPDATE tavsiye
SET degisken = (select urun_ad from urunler where urun_ID=@en_cok_satan_urun)
WHERE Messages_Id = 8;
```

-- bu hafta en az satan ürün

```
SELECT TOP 1 @bu_hafta_en_az_satan_urun = urun_Id
FROM dbo.satis
WHERE tarih >= DATEADD(week, -1, GETDATE())
GROUP BY urun_Id
ORDER BY (SUM(adet) / (SELECT SUM(adet) FROM dbo.satis WHERE tarih < DATEADD(week, -1, GETDATE()))) * 100);
```

--update

```
UPDATE tavsiye
SET degisken = (select urun_ad from urunler where urun_ID=@en_az_satan_urun)
WHERE Messages_Id = 9;
```

-- son 2 hafta satılmayan ürün\_id

```
SELECT @sayi=(count(urun_ID))
FROM dbo.satis
WHERE tarih >= DATEADD(week, -2, GETDATE())
GROUP BY urun_Id
```

```
HAVING SUM(adet) = 0;
```

```
SELECT @son_iki_hafta_satilmayan_urunler = STUFF((  
    SELECT ', ' + CAST(u.urun_ad AS VARCHAR(255))  
    FROM dbo.satis s  
    INNER JOIN dbo.urunler u ON s.urun_ID = u.urun_ID  
    WHERE s.tarih <= DATEADD(WEEK, -2, GETDATE())  
    GROUP BY u.urun_ad  
    FOR XML PATH('')  
), 1, 2, '')
```

```
SET @son_iki_hafta_satilmayan_urunler = COALESCE(@son_iki_hafta_satilmayan_urunler + ', ', '')
```

```
--update
```

```
UPDATE tavsiye
```

```
SET degisken = @son_iki_hafta_satilmayan_urunler
```

```
WHERE Messages_Id = 1;
```

```
end
```

```
--en çok satan ürün --
--en az satan ürün --
--en çok satan kategori --
--en az satan kategori --
--en çok satan marka --
--en az satan marka --
--bu hafta en çok satış oranı olan ürün (geçen haftaki satış oranı/bu haftaki satış oranı*100) --
--bu hafta en az satış oranı olan ürün (geçen haftaki satış oranı/bu haftaki satış oranı*100) --
--son 2 hafta bu üründen satmadın --
-- gibi mesajların hazır bulunduğu bir tablo yapısı vardır her insert işlemi yapıldığında--
--tablo yapısı urun_Id,tarih ve adet bilgisinin bulunduğu bir satış tablosunda çalışacak bir trigger --
--yazmak istenmektedir bu trigger çalışınca tavsiyeler tablosundaki bulunun urun_Id ler update
olmalı--
--tavsiye tablosu: messages_Id,urun_ID,urun_tavsiyesi--
GO
```

```
ALTER TABLE [dbo].[satis] ENABLE TRIGGER [tavsiye_Trigger]
```

```
GO
```

```
USE [AH_firmasi]
```

```
GO
```

```
/***** Object: Trigger [dbo].[iss] Script Date: 26.05.2023 04:05:42 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TRIGGER [dbo].[iss]
```

```
ON [dbo].[siparisler]
```

```
AFTER UPDATE
```



AS

BEGIN

declare @stok int=(select top 1 iss from urunler where urun\_ID=(select top 1 urun\_ID from inserted))

declare @durum1 int=(select top 1 durum\_id from inserted)

declare @durum2 int=(select top 1 durum\_id from deleted)

if(@durum1=2)

begin

update urunler set iss=(iss-(select top 1 adet from inserted)) where urun\_ID=(select top 1 urun\_ID from inserted)

end

else if(@durum2=2 and @durum1=0)

begin

update urunler set iss=(iss+(select top 1 adet from inserted))where urun\_ID=(select top 1 urun\_ID from inserted)

end

END

GO

ALTER TABLE [dbo].[siparisler] ENABLE TRIGGER [iss]

GO

USE [AH\_firmasi]

GO

/\*\*\*\*\* Object: Trigger [dbo].[satisfiyat] Script Date: 26.05.2023 04:05:50 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

```

CREATE TRIGGER [dbo].[satisfiyat]
ON [dbo].[siparisler]
AFTER INSERT
AS
BEGIN
    DECLARE @fiyat INT
declare @maliyet int

    SELECT TOP 1 @fiyat = satis,@maliyet=maliyet FROM urunler WHERE urun_ID IN (SELECT top 1
urun_ID FROM inserted)

    UPDATE siparisler SET anlik_fiyat = @fiyat,anlik_maliyet=@maliyet WHERE siparis_id IN (SELECT
TOP 1 siparis_id FROM inserted)
END
GO

```

```

ALTER TABLE [dbo].[siparisler] ENABLE TRIGGER [satisfiyat]
GO
USE [AH_firmasi]
GO

```

```

/***** Object: Trigger [dbo].[sepet1]  Script Date: 26.05.2023 04:05:58 *****/
SET ANSI_NULLS ON
GO

```

```

SET QUOTED_IDENTIFIER ON
GO

```

```

CREATE TRIGGER [dbo].[sepet1]
ON [dbo].[siparisler]
AFTER insert
as
BEGIN
declare @adet int;

```

```
declare @id int;

declare @date date;

set @adet=(select adet from inserted where durum_id=4);

set @id=(select urun_ID from inserted where durum_id=4);

set @date=(select siparis_tarih from inserted where durum_id=4);

insert into sepet values(@id,@date,@adet);

end

GO
```

```
ALTER TABLE [dbo].[siparisler] ENABLE TRIGGER [sepet1]
```

```
GO
```

```
USE [AH_firmasi]
```

```
GO
```

```
/***** Object: Trigger [dbo].[siparis1]  Script Date: 26.05.2023 04:06:07 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TRIGGER [dbo].[siparis1]
```

```
ON [dbo].[siparisler]
```

```
AFTER UPDATE
```

```
as
```

```
BEGIN
```

```
declare @adet int;
```

```
declare @genel_adet int;
```

```
declare @urun_id int;
```

```
declare @durum_yeni int;
```

```
declare @anlik_fiyat int;
```

```

declare @magaza_id int;

declare @tarih datetime;

declare @siparis_id int;

declare @anlik_maliyet int;

        set @siparis_id = (select siparis_id from inserted);

        set @anlik_fiyat = (select anlik_fiyat from inserted);

        set @anlik_maliyet = (select anlik_maliyet from inserted);

        set @durum_yeni = (select durum_id from inserted);

set @urun_id=(select urun_ID from inserted);

        set @adet = (select adet from inserted);


select @genel_adet=urun_adet from urunler where urun_ID=@urun_id

select @tarih=siparis_tarih from inserted;

set @magaza_id = (select magaza_id from inserted);


if(@durum_yeni=3)

begin

insert into satis values (@urun_id,CURRENT_TIMESTAMP,(@adet),@magaza_id,@tarih);

update urunler set urun_adet=urun_adet-@adet where urun_ID=@urun_id;

insert into gecmis
values(@urun_id,CURRENT_TIMESTAMP,@adet,@anlik_fiyat,@magaza_id,@tarih,@durum_yeni,@
anlik_maliyet);

delete from siparisler where siparis_id=@siparis_id;

end

else if(@durum_yeni=0)

begin

insert into gecmis
values(@urun_id,CURRENT_TIMESTAMP,@adet,@anlik_fiyat,@magaza_id,@tarih,@durum_yeni,@
anlik_maliyet);

delete from siparisler where siparis_id=@siparis_id;

end

```

END

GO

ALTER TABLE [dbo].[siparisler] ENABLE TRIGGER [siparis1]

GO

USE [AH\_firmasi]

GO

/\*\*\*\*\* Object: Trigger [dbo].[alis\_satisk] Script Date: 26.05.2023 04:06:36 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TRIGGER [dbo].[alis\_satisk]

ON [dbo].[urunler]

AFTER INSERT,DELETE,UPDATE

as

BEGIN

declare @yeni int;

declare @eski int;

declare @id int;

set @yeni = (select top 1 urun\_adet from inserted);

set @eski = (select top 1 urun\_adet from deleted);

set @id=(select top 1 urun\_ID from inserted);

```
if(@yeni>=0)begin
    if(@yeni>@eski)--alış
        insert into alis values (@id,CURRENT_TIMESTAMP,(@yeni-@eski));
    else if (@yeni=@eski)--alış
        declare @aytacyapti int
    else
        insert into alis values (@id,CURRENT_TIMESTAMP,(@yeni));
    end
    else
        rollback;
END
GO
```

```
ALTER TABLE [dbo].[urunler] ENABLE TRIGGER [alis_satisk]
GO
USE [AH_firmasi]
GO
```

```
/***** Object: Trigger [dbo].[sifirkontrol]  Script Date: 26.05.2023 04:06:52 *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
create trigger [dbo].[sifirkontrol] on [dbo].[urunler]
after update,insert,delete
as begin
    declare @adet int;
    select @adet=urun_adet from urunler
```

```
if @adet<=-1
```

```
rollback;
```

```
end
```

```
GO
```

```
ALTER TABLE [dbo].[urunler] ENABLE TRIGGER [sifirkontrol]
```

```
GO
```

```
USE [AH_firmasi]
```

```
GO
```

```
/***** Object: Trigger [dbo].[tavsiye_Trigger2] Script Date: 26.05.2023 04:07:05 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TRIGGER [dbo].[tavsiye_Trigger2]
```

```
ON [dbo].[urunler]
```

```
AFTER INSERT,update,delete
```

```
AS
```

```
BEGIN
```

```
DECLARE @en_cok_satan_urun INT;
```

```
DECLARE @en_az_satan_urun INT;
```

```
DECLARE @en_cok_satan_kategori INT;
```

```
DECLARE @en_az_satan_kategori INT;
```

```
DECLARE @en_cok_satan_marka INT;
```

```
DECLARE @en_az_satan_marka INT;
```

```
DECLARE @en_cok_satan_urun_oran INT;
```

```
DECLARE @en_az_satan_urun_oran INT;
```

--bu gün

DECLARE @en\_cok\_satan\_urun1 INT;

DECLARE @en\_az\_satan\_urun1 INT;

DECLARE @en\_cok\_satan\_kategori1 INT;

DECLARE @en\_az\_satan\_kategori1 INT;

DECLARE @en\_cok\_satan\_marka1 INT;

DECLARE @en\_az\_satan\_marka1 INT;

DECLARE @en\_cok\_satan\_urun\_oran1 INT;

DECLARE @en\_az\_satan\_urun\_oran1 INT;

--bu hafta

DECLARE @en\_cok\_satan\_urun2 INT;

DECLARE @en\_az\_satan\_urun2 INT;

DECLARE @en\_cok\_satan\_kategori2 INT;

DECLARE @en\_az\_satan\_kategori2 INT;

DECLARE @en\_cok\_satan\_marka2 INT;

DECLARE @en\_az\_satan\_marka2 INT;

DECLARE @en\_cok\_satan\_urun\_oran2 INT;

DECLARE @en\_az\_satan\_urun\_oran2 INT;

--bu ay

DECLARE @en\_cok\_satan\_urun3 INT;

DECLARE @en\_az\_satan\_urun3 INT;

DECLARE @en\_cok\_satan\_kategori3 INT;

DECLARE @en\_az\_satan\_kategori3 INT;

DECLARE @en\_cok\_satan\_marka3 INT;

DECLARE @en\_az\_satan\_marka3 INT;

DECLARE @en\_cok\_satan\_urun\_oran3 INT;

DECLARE @en\_az\_satan\_urun\_oran3 INT;

--bu yıl



```
DECLARE @son_iki_hafta_satilmayan_urunler varchar(100);  
declare @sayi int;
```

```
-----bu gün-----  
-----
```

```
-- en çok satan ürün  
SELECT TOP 1 @en_cok_satan_urun = urun_ID  
FROM dbo.satis  
where tarih=CURRENT_TIMESTAMP  
GROUP BY urun_Id  
ORDER BY SUM(adet) DESC;  
  
--update  
IF @en_cok_satan_urun IS NOT NULL  
UPDATE tavsiye  
SET degisken = (SELECT urun_ad FROM urunler WHERE urun_ID = @en_cok_satan_urun)  
WHERE Messages_Id = 2;  
else  
UPDATE tavsiye  
SET degisken = 'bu gün satış olmamıştır'  
WHERE Messages_Id = 2;
```

```
-- en az satan ürün  
SELECT TOP 1 @en_az_satan_urun = urun_Id  
FROM dbo.satis  
where tarih=CURRENT_TIMESTAMP  
GROUP BY urun_Id  
ORDER BY SUM(adet);
```

```
IF @en_az_satan_urun IS NOT NULL

BEGIN

    UPDATE tavsiye

    SET degisken = (SELECT urun_ad FROM urunler WHERE urun_ID = @en_az_satan_urun)

    WHERE Messages_Id = 3;

END;

else

UPDATE tavsiye

SET degisken = 'bu gün satış olmamıştır'

WHERE Messages_Id = 3;


-- en çok satan kategori

SELECT TOP 1 @en_cok_satan_kategori = kategori_ID

FROM dbo.urunler

INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id

where tarih=CURRENT_TIMESTAMP

GROUP BY kategori_ID

ORDER BY SUM(adet) DESC;


--update


IF @en_cok_satan_kategori IS NOT NULL

BEGIN

    UPDATE tavsiye

    SET degisken = (SELECT urun_ad FROM urunler WHERE urun_ID = @en_cok_satan_urun)

    WHERE Messages_Id = 4;

END;

else

UPDATE tavsiye

SET degisken = 'bu gün satış olmamıştır'

WHERE Messages_Id = 4;
```

```
-- en az satan kategori

SELECT TOP 1 @en_az_satan_kategori = kategori_ID
FROM dbo.urunler
INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id
where tarih=CURRENT_TIMESTAMP
GROUP BY kategori_ID
ORDER BY SUM(adet);

--update
IF @en_az_satan_kategori IS NOT NULL
BEGIN
    UPDATE tavsiye
    SET degisken = (SELECT urun_ad FROM urunler WHERE urun_ID = @en_az_satan_urun)
    WHERE Messages_Id = 5;
END;
else
UPDATE tavsiye
SET degisken ='bu gün satış olmamıştır'
WHERE Messages_Id = 5;

-- en çok satan marka

SELECT TOP 1 @en_cok_satan_marka = marka_ID
FROM dbo.urunler
INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id
where tarih=CURRENT_TIMESTAMP
GROUP BY marka_ID
ORDER BY SUM(adet) DESC;

--update
IF @en_cok_satan_marka IS NOT NULL
BEGIN
```

```
UPDATE tavsiye
SET degisken = (SELECT urun_ad FROM urunler WHERE urun_ID = @en_cok_satan_urun)
WHERE Messages_Id = 6;
END;
else
UPDATE tavsiye
SET degisken = 'bu gün satış olmamıştır'
WHERE Messages_Id = 6;

-- en az satan marka
SELECT TOP 1 @en_az_satan_marka = marka_ID
FROM dbo.urunler
INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id
where tarih=CURRENT_TIMESTAMP
GROUP BY marka_ID
ORDER BY SUM(adet);

--update
IF @en_az_satan_marka IS NOT NULL
BEGIN
    UPDATE tavsiye
    SET degisken = (SELECT urun_ad FROM urunler WHERE urun_ID = @en_az_satan_urun)
    WHERE Messages_Id = 7;
END;
else
UPDATE tavsiye
SET degisken = 'bu gün satış olmamıştır'
WHERE Messages_Id = 7;

-- bu gün en çok satan ürün
SELECT TOP 1 @en_cok_satan_urun_oran = urun_Id
```

```
FROM dbo.satis

where tarih=CURRENT_TIMESTAMP

GROUP BY urun_Id

ORDER BY (SUM(adet) / (SELECT SUM(adet) FROM dbo.satis WHERE tarih < DATEADD(week, -1,
GETDATE())) * 100) DESC;
```

--update

```
IF @en_cok_satan_urun_oran IS NOT NULL
```

```
BEGIN
```

```
UPDATE tavsiye
```

```
SET degisken = (SELECT urun_ad FROM urunler WHERE urun_ID = @en_cok_satan_urun)
```

```
WHERE Messages_Id = 8;
```

```
END;
```

```
else
```

```
UPDATE tavsiye
```

```
SET degisken = 'bu gün satis olmamıştır'
```

```
WHERE Messages_Id = 8;
```

-- bu gün en az satan ürün

```
SELECT TOP 1 @en_az_satan_urun_oran = urun_Id
```

```
FROM dbo.satis
```

```
where tarih=CURRENT_TIMESTAMP
```

```
GROUP BY urun_Id
```

```
ORDER BY (SUM(adet) / (SELECT SUM(adet) FROM dbo.satis WHERE tarih < DATEADD(week, -1,
GETDATE())) * 100);
```

--update

```
IF @en_az_satan_urun_oran IS NOT NULL
```

```
BEGIN
```

```
UPDATE tavsiye
```

```
SET degisken = (SELECT urun_ad FROM urunler WHERE urun_ID = @en_az_satan_urun)
```

```
WHERE Messages_Id = 9;
```

END;

else

UPDATE tavsiye

SET degisken ='bu gün satış olmamıştır'

WHERE Messages\_Id = 9;

-----bu gün bitis-----  
-----

-- en çok satan ürün

SELECT TOP 1 @en\_cok\_satan\_urun1 = urun\_ID

FROM dbo.satis

WHERE DATEPART(wk, [tarih]) = DATEPART(wk, CURRENT\_TIMESTAMP) AND YEAR([tarih]) =  
YEAR(CURRENT\_TIMESTAMP)

GROUP BY urun\_Id

ORDER BY SUM(adet) DESC;

--update

IF @en\_cok\_satan\_urun1 IS NOT NULL

UPDATE tavsiye

SET degisken = (select urun\_ad from urunler where urun\_ID=@en\_cok\_satan\_urun1)

WHERE Messages\_Id = 10;

else

UPDATE tavsiye

SET degisken ='bu hafta satış olmamıştır'

WHERE Messages\_Id = 10;

-- en az satan ürün

SELECT TOP 1 @en\_az\_satan\_urun1 = urun\_Id

FROM dbo.satis

```
WHERE DATEPART(wk, [tarih]) = DATEPART(wk, CURRENT_TIMESTAMP) AND YEAR([tarih]) =  
YEAR(CURRENT_TIMESTAMP)
```

```
GROUP BY urun_Id
```

```
ORDER BY SUM(adet);
```

```
--update
```

```
IF @en_az_satan_urun1 IS NOT NULL
```

```
UPDATE tavsiye
```

```
SET degisken = (select urun_ad from urunler where urun_ID=@en_az_satan_urun1)
```

```
WHERE Messages_Id = 11;
```

```
else
```

```
UPDATE tavsiye
```

```
SET degisken = 'bu hafta satis olmamıştır'
```

```
WHERE Messages_Id = 11;
```

```
-- en çok satan kategori
```

```
SELECT TOP 1 @en_cok_satan_kategori1 = kategori_ID
```

```
FROM dbo.urunler
```

```
INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id
```

```
WHERE DATEPART(wk, [tarih]) = DATEPART(wk, CURRENT_TIMESTAMP) AND YEAR([tarih]) =  
YEAR(CURRENT_TIMESTAMP)
```

```
GROUP BY kategori_ID
```

```
ORDER BY SUM(adet) DESC;
```

```
--update
```

```
IF @en_cok_satan_kategori1 IS NOT NULL
```

```
UPDATE tavsiye
```

```
SET degisken = (select kategori_AD from kategoriler where kategori_ID=@en_cok_satan_kategori1)
```

```
WHERE Messages_Id = 12;
```

```
else
```

```
UPDATE tavsiye
```

```
SET degisken = 'bu hafta satis olmamıştır'
```

```
WHERE Messages_Id = 12;
```

```
-- en az satan kategori
```

```
SELECT TOP 1 @en_az_satan_kategori1 = kategori_ID
```

```
FROM dbo.urunler
```

```
INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id
```

```
WHERE DATEPART(wk, [tarih]) = DATEPART(wk, CURRENT_TIMESTAMP) AND YEAR([tarih]) =  
YEAR(CURRENT_TIMESTAMP)
```

```
GROUP BY kategori_ID
```

```
ORDER BY SUM(adet);
```

```
--update
```

```
IF @en_cok_satan_kategori1 IS NOT NULL
```

```
UPDATE tavsiye
```

```
SET degisken = (select kategori_AD from kategoriler where kategori_ID=@en_az_satan_kategori1)
```

```
WHERE Messages_Id = 13;
```

```
else
```

```
UPDATE tavsiye
```

```
SET degisken = 'bu hafta satis olmamıştır'
```

```
WHERE Messages_Id = 13;
```

```
-- en çok satan marka
```

```
SELECT TOP 1 @en_cok_satan_marka1 = marka_ID
```

```
FROM dbo.urunler
```

```
INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id
```

```
WHERE DATEPART(wk, [tarih]) = DATEPART(wk, CURRENT_TIMESTAMP) AND YEAR([tarih]) =  
YEAR(CURRENT_TIMESTAMP)
```

```
GROUP BY marka_ID
```

```
ORDER BY SUM(adet) DESC;
```

```
--update
```

```
IF @en_cok_satan_marka1 IS NOT NULL
```

```
UPDATE tavsiye
```



```

SET degisken = (select marka_AD from markalar where marka_ID=@en_cok_satan_marka1)

WHERE Messages_Id = 14;

else

UPDATE tavsiye

SET degisken ='bu hafta satis olmamıştır'

WHERE Messages_Id = 14;

-- en az satan marka

SELECT TOP 1 @en_az_satan_marka1 = marka_ID

FROM dbo.urunler

INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id

WHERE DATEPART(wk, [tarih]) = DATEPART(wk, CURRENT_TIMESTAMP) AND YEAR([tarih]) =
YEAR(CURRENT_TIMESTAMP)

GROUP BY marka_ID

ORDER BY SUM(adet);

--update

IF @en_cok_satan_marka1 IS NOT NULL

UPDATE tavsiye

SET degisken = (select marka_AD from markalar where marka_ID=@en_az_satan_marka1)

WHERE Messages_Id = 15;

else

UPDATE tavsiye

SET degisken ='bu hafta satis olmamıştır'

WHERE Messages_Id = 15;

-- bu gün en çok satan ürün

SELECT TOP 1 @en_cok_satan_urun_oran1 = urun_Id

FROM dbo.satis

WHERE DATEPART(wk, [tarih]) = DATEPART(wk, CURRENT_TIMESTAMP) AND YEAR([tarih]) =
YEAR(CURRENT_TIMESTAMP)

GROUP BY urun_Id

```

```
ORDER BY (SUM(adet) / (SELECT SUM(adet) FROM dbo.satis WHERE tarih < DATEADD(week, -1,
GETDATE())) * 100) DESC;
```

```
--update
```

```
IF @en_cok_satan_urun1 IS NOT NULL
```

```
UPDATE tavsiye
```

```
SET degisken = (select urun_ad from urunler where urun_ID=@en_cok_satan_urun1)
```

```
WHERE Messages_Id = 16;
```

```
else
```

```
UPDATE tavsiye
```

```
SET degisken = 'bu hafta satis olmamıştır'
```

```
WHERE Messages_Id = 16;
```

```
-- bu gün en az satan ürün
```

```
SELECT TOP 1 @en_az_satan_urun_oran1 = urun_Id
```

```
FROM dbo.satis
```

```
WHERE DATEPART(wk, [tarih]) = DATEPART(wk, CURRENT_TIMESTAMP) AND YEAR([tarih]) =
YEAR(CURRENT_TIMESTAMP)
```

```
GROUP BY urun_Id
```

```
ORDER BY (SUM(adet) / (SELECT SUM(adet) FROM dbo.satis WHERE tarih < DATEADD(week, -1,
GETDATE())) * 100);
```

```
--update
```

```
IF @en_az_satan_urun1 IS NOT NULL
```

```
UPDATE tavsiye
```

```
SET degisken = (select urun_ad from urunler where urun_ID=@en_az_satan_urun1)
```

```
WHERE Messages_Id = 17;
```

```
else
```

```
UPDATE tavsiye
```

```
SET degisken = 'bu hafta satis olmamıştır'
```

```
WHERE Messages_Id = 17;
```

```
-----bu hafta bitis-----
-----
```

-----bu ay-----  
-----

-- en çok satan ürün

SELECT TOP 1 @en\_cok\_satan\_urun2 = urun\_ID

FROM dbo.satis

WHERE MONTH([tarih]) = MONTH(CURRENT\_TIMESTAMP) AND YEAR([tarih]) =  
YEAR(CURRENT\_TIMESTAMP)

GROUP BY urun\_Id

ORDER BY SUM(adet) DESC;

--update

IF @en\_cok\_satan\_urun2 IS NOT NULL

UPDATE tavsiye

SET degisken = (select urun\_ad from urunler where urun\_ID=@en\_cok\_satan\_urun2)

WHERE Messages\_Id = 18;

else

UPDATE tavsiye

SET degisken = 'bu ay satis olmamıştır'

WHERE Messages\_Id = 18;

-- en az satan ürün

SELECT TOP 1 @en\_az\_satan\_urun2 = urun\_Id

FROM dbo.satis

WHERE MONTH([tarih]) = MONTH(CURRENT\_TIMESTAMP) AND YEAR([tarih]) =  
YEAR(CURRENT\_TIMESTAMP)

GROUP BY urun\_Id

ORDER BY SUM(adet);

--update

IF @en\_az\_satan\_urun2 IS NOT NULL

UPDATE tavsiye

```

SET degisken = (select urun_ad from urunler where urun_ID=@en_az_satan_urun2)

WHERE Messages_Id = 19;

else

UPDATE tavsiye

SET degisken ='bu ay satis olmamıştır'

WHERE Messages_Id = 19;

-- en çok satan kategori

SELECT TOP 1 @en_cok_satan_kategori2 = kategori_ID

FROM dbo.urunler

INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id

WHERE MONTH([tarih]) = MONTH(CURRENT_TIMESTAMP) AND YEAR([tarih]) =
YEAR(CURRENT_TIMESTAMP)

GROUP BY kategori_ID

ORDER BY SUM(adet) DESC;

--update

IF @en_cok_satan_urun2 IS NOT NULL

UPDATE tavsiye

SET degisken = (select kategori_AD from kategoriler where kategori_ID=@en_cok_satan_kategori2)

WHERE Messages_Id = 20;

else

UPDATE tavsiye

SET degisken ='bu ay satis olmamıştır'

WHERE Messages_Id = 20;

-- en az satan kategori

SELECT TOP 1 @en_az_satan_kategori2 = kategori_ID

FROM dbo.urunler

INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id

WHERE MONTH([tarih]) = MONTH(CURRENT_TIMESTAMP) AND YEAR([tarih]) =
YEAR(CURRENT_TIMESTAMP)

GROUP BY kategori_ID

```

```

ORDER BY SUM(adet);

--update

IF @en_az_satan_kategori2 IS NOT NULL

UPDATE tavsiye

SET degisken = (select kategori_AD from kategoriler where kategori_ID=@en_az_satan_kategori2)

WHERE Messages_Id = 21;

else

UPDATE tavsiye

SET degisken = 'bu ay satis olmamıştır'

WHERE Messages_Id = 21;

-- en çok satan marka

SELECT TOP 1 @en_cok_satan_marka2 = marka_ID

FROM dbo.urunler

INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id

WHERE MONTH([tarih]) = MONTH(CURRENT_TIMESTAMP) AND YEAR([tarih]) =

YEAR(CURRENT_TIMESTAMP)

GROUP BY marka_ID

ORDER BY SUM(adet) DESC;

--update

IF @en_cok_satan_marka2 IS NOT NULL

UPDATE tavsiye

SET degisken = (select marka_AD from markalar where marka_ID=@en_cok_satan_marka2)

WHERE Messages_Id = 22;

else

UPDATE tavsiye

SET degisken = 'bu ay satis olmamıştır'

WHERE Messages_Id = 22;

-- en az satan marka

SELECT TOP 1 @en_az_satan_marka2 = marka_ID

FROM dbo.urunler

```

```

INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id

WHERE MONTH([tarih]) = MONTH(CURRENT_TIMESTAMP) AND YEAR([tarih]) =
YEAR(CURRENT_TIMESTAMP)

GROUP BY marka_Id

ORDER BY SUM(adet);

--update

IF @en_az_satan_marka2 IS NOT NULL

UPDATE tavsiye

SET degisken = (select marka_AD from markalar where marka_Id=@en_az_satan_marka2)

WHERE Messages_Id = 23;

else

UPDATE tavsiye

SET degisken = 'bu ay satis olmamıştır'

WHERE Messages_Id = 23;

-- bu gün en çok satan ürün

SELECT TOP 1 @en_cok_satan_urun_oran2 = urun_Id

FROM dbo.satis

WHERE MONTH([tarih]) = MONTH(CURRENT_TIMESTAMP) AND YEAR([tarih]) =
YEAR(CURRENT_TIMESTAMP)

GROUP BY urun_Id

ORDER BY (SUM(adet) / (SELECT SUM(adet) FROM dbo.satis WHERE tarih < DATEADD(week, -1,
GETDATE())) * 100) DESC;

--update

IF @en_cok_satan_urun2 IS NOT NULL

UPDATE tavsiye

SET degisken = (select urun_ad from urunler where urun_Id=@en_cok_satan_urun2)

WHERE Messages_Id = 24;

else

UPDATE tavsiye

```

SET degisken ='bu ay satis olmamıştır'

WHERE Messages\_Id = 24;

-- bu gün en az satan ürün

SELECT TOP 1 @en\_az\_satan\_urun\_oran2 = urun\_Id

FROM dbo.satis

WHERE MONTH([tarih]) = MONTH(CURRENT\_TIMESTAMP) AND YEAR([tarih]) =  
YEAR(CURRENT\_TIMESTAMP)

GROUP BY urun\_Id

ORDER BY (SUM(adet) / (SELECT SUM(adet) FROM dbo.satis WHERE tarih < DATEADD(week, -1,  
GETDATE())) \* 100);

--update

IF @en\_az\_satan\_urun2 IS NOT NULL

UPDATE tavsiye

SET degisken = (select urun\_ad from urunler where urun\_ID=@en\_az\_satan\_urun2)

WHERE Messages\_Id = 25;

else

UPDATE tavsiye

SET degisken ='bu ay satis olmamıştır'

WHERE Messages\_Id = 25;

-----bu ay bitis-----  
-----

-----bu yıl-----  
-----

-- en çok satan ürün

SELECT TOP 1 @en\_cok\_satan\_urun3 = urun\_ID

FROM dbo.satis

WHERE YEAR([tarih]) = YEAR(CURRENT\_TIMESTAMP)

GROUP BY urun\_Id

ORDER BY SUM(adet) DESC;

--update

UPDATE tavsiye

SET degisken = (select urun\_ad from urunler where urun\_ID=@en\_cok\_satan\_urun3)

WHERE Messages\_Id = 26;

-- en az satan ürün

SELECT TOP 1 @en\_az\_satan\_urun3 = urun\_Id

FROM dbo.satis

WHERE YEAR([tarih]) = YEAR(CURRENT\_TIMESTAMP)

GROUP BY urun\_Id

ORDER BY SUM(adet);

--update

UPDATE tavsiye

SET degisken = (select urun\_ad from urunler where urun\_ID=@en\_az\_satan\_urun3)

WHERE Messages\_Id = 27;

-- en çok satan kategori

SELECT TOP 1 @en\_cok\_satan\_kategori3 = kategori\_ID

FROM dbo.urunler

INNER JOIN dbo.satis ON dbo.urunler.urun\_Id = dbo.satis.urun\_Id

WHERE YEAR([tarih]) = YEAR(CURRENT\_TIMESTAMP)

GROUP BY kategori\_ID

ORDER BY SUM(adet) DESC;

--update

UPDATE tavsiye

SET degisken = (select kategori\_AD from kategoriler where kategori\_ID=@en\_cok\_satan\_kategori3)

WHERE Messages\_Id = 28;



```
-- en az satan kategori

SELECT TOP 1 @en_az_satan_kategori3 = kategori_ID
FROM dbo.urunler

INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id
WHERE YEAR([tarih]) = YEAR(CURRENT_TIMESTAMP)

GROUP BY kategori_ID

ORDER BY SUM(adet);

--update

UPDATE tavsiye

SET degisken = (select kategori_AD from kategoriler where kategori_ID=@en_az_satan_kategori3)
WHERE Messages_Id = 29;
```

```
-- en çok satan marka

SELECT TOP 1 @en_cok_satan_marka3 = marka_ID
FROM dbo.urunler

INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id
WHERE YEAR([tarih]) = YEAR(CURRENT_TIMESTAMP)

GROUP BY marka_ID

ORDER BY SUM(adet) DESC;
```

```
--update

UPDATE tavsiye

SET degisken = (select marka_AD from markalar where marka_ID=@en_cok_satan_marka3)
WHERE Messages_Id = 30;
```

```
-- en az satan marka

SELECT TOP 1 @en_az_satan_marka3 = marka_ID
FROM dbo.urunler

INNER JOIN dbo.satis ON dbo.urunler.urun_Id = dbo.satis.urun_Id
WHERE YEAR([tarih]) = YEAR(CURRENT_TIMESTAMP)
```

```
GROUP BY marka_ID  
ORDER BY SUM(adet);
```

```
--update
```

```
UPDATE tavsiye
```

```
SET degisken = (select marka_AD from markalar where marka_ID=@en_az_satan_marka3)
```

```
WHERE Messages_Id = 31;
```

```
-- bu gün en çok satan ürün
```

```
SELECT TOP 1 @en_cok_satan_urun_oran3 = urun_Id
```

```
FROM dbo.satis
```

```
WHERE YEAR([tarih]) = YEAR(CURRENT_TIMESTAMP)
```

```
GROUP BY urun_Id
```

```
ORDER BY (SUM(adet) / (SELECT SUM(adet) FROM dbo.satis WHERE tarih < DATEADD(week, -1,  
GETDATE())) * 100) DESC;
```

```
--update
```

```
UPDATE tavsiye
```

```
SET degisken = (select urun_ad from urunler where urun_ID=@en_cok_satan_urun3)
```

```
WHERE Messages_Id = 32;
```

```
-- bu gün en az satan ürün
```

```
SELECT TOP 1 @en_az_satan_urun_oran3 = urun_Id
```

```
FROM dbo.satis
```

```
WHERE YEAR([tarih]) = YEAR(CURRENT_TIMESTAMP)
```

```
GROUP BY urun_Id
```

```
ORDER BY (SUM(adet) / (SELECT SUM(adet) FROM dbo.satis WHERE tarih < DATEADD(week, -1,  
GETDATE())) * 100);
```

```
--update
```

```
UPDATE tavsiye
```

```
SET degisken = (select urun_ad from urunler where urun_ID=@en_az_satan_urun3)
```

```
WHERE Messages_Id = 33;
```

```
-----bu yıl bitis-----  
-----
```

```
-- son 2 hafta satılmayan ürün_id
```

```
IF EXISTS (
```

```
    SELECT 1
```

```
    FROM dbo.satis s
```

```
    WHERE s.tarih >= DATEADD(WEEK, -2, GETDATE())
```

```
    GROUP BY s.urun_ID
```

```
    HAVING SUM(s.adet) = 0
```

```
)
```

```
BEGIN
```

```
    SELECT @son_iki_hafta_satilmayan_urunler = STUFF((
```

```
        SELECT ' ' + CAST(u.urun_ad AS VARCHAR(255))
```

```
        FROM dbo.satis s
```

```
        INNER JOIN dbo.urunler u ON s.urun_ID = u.urun_ID
```

```
        WHERE s.tarih <= DATEADD(WEEK, -2, GETDATE())
```

```
        GROUP BY u.urun_ad
```

```
        FOR XML PATH('')
```

```
    ), 1, 2, '')
```

```
SET @son_iki_hafta_satilmayan_urunler = COALESCE(@son_iki_hafta_satilmayan_urunler + ' ', '')
```

```
--update
```

```
UPDATE tavsiye
```

```
SET degisken = @son_iki_hafta_satilmayan_urunler
```

```
WHERE Messages_Id = 1;
```

END

ELSE

BEGIN

UPDATE tavsiye

SET degisken = 'yoktur'

WHERE Messages\_Id = 1;

END

end

GO

ALTER TABLE [dbo].[urunler] ENABLE TRIGGER [tavsiye\_Trigger2]

GO

USE [AH\_firmasi]

GO

/\*\*\*\*\* Object: Trigger [dbo].[urun\_adet\_iss\_kontrol] Script Date: 26.05.2023 04:07:30 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TRIGGER [dbo].[urun\_adet\_iss\_kontrol]

ON [dbo].[urunler]

FOR INSERT,update

AS

BEGIN

```
if(((select iss from inserted)is null))
begin
declare @urunadet int;
set @urunadet =(select urun_adet from inserted)
UPDATE urunler
    SET iss = @urunadet
where urun_ID=(select top 1 urun_ID from inserted)
end
else if((select urun_adet from inserted)>(select urun_adet from deleted))
begin
UPDATE urunler
SET iss = iss + ((select urun_adet from inserted) - (select urun_adet from deleted))
WHERE urun_ID = (select top 1 urun_ID from inserted)

end

END
GO
```

```
ALTER TABLE [dbo].[urunler] ENABLE TRIGGER [urun_adet_iss_kontrol]
GO
```

AH\_firması database kısmı bitmiştir

LoginDB database sql kodları:

Tablolar

```
USE [LoginDB]
GO
```

/\*\*\*\*\* Object: Table [dbo].[lisanslar] Script Date: 26.05.2023 04:10:48 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[lisanslar](

[Is] [int] NOT NULL,

[lisans\_adlari] [varchar](50) NULL,

CONSTRAINT [PK\_lisanslar] PRIMARY KEY CLUSTERED

(

[Is] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF,  
ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF)  
ON [PRIMARY]

) ON [PRIMARY]

GO

USE [LoginDB]

GO

/\*\*\*\*\* Object: Table [dbo].[sirketad] Script Date: 26.05.2023 04:10:59 \*\*\*\*\*/

SET ANSI\_NULLS ON

GO

SET QUOTED\_IDENTIFIER ON

GO

CREATE TABLE [dbo].[sirketad](

[sirket\_ID] [int] NOT NULL,

[Sirket\_AD] [varchar](50) NULL,

CONSTRAINT [PK\_sirketad] PRIMARY KEY CLUSTERED

```
(  
[sirket_ID] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)  
ON [PRIMARY]  
) ON [PRIMARY]  
GO
```

```
USE [LoginDB]  
GO
```

```
/***** Object: Table [dbo].[Users]  Script Date: 26.05.2023 04:11:37 *****/  
SET ANSI_NULLS ON  
GO
```

```
SET QUOTED_IDENTIFIER ON  
GO
```

```
CREATE TABLE [dbo].[Users](  
[id] [int] IDENTITY(1,1) NOT NULL,  
[firstLast] [nvarchar](max) NULL,  
[email] [nvarchar](max) NULL,  
[sk] [varchar](50) NOT NULL,  
[ls] [int] NOT NULL,  
[pass] [nvarchar](max) NULL,  
[since] [date] NULL,  
[sirket_ID] [int] NULL,  
[dakika_1gun] [int] NULL,  
[dakika_7gun] [int] NULL,  
CONSTRAINT [PK_Users] PRIMARY KEY CLUSTERED  
(  
[id] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,  
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON, OPTIMIZE_FOR_SEQUENTIAL_KEY = OFF)  
ON [PRIMARY]
```

```
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
```

```
GO
```

```
ALTER TABLE [dbo].[Users] WITH CHECK ADD CONSTRAINT [FK_Users_lisanslar] FOREIGN KEY([ls])  
REFERENCES [dbo].[lisanslar] ([ls])
```

```
GO
```

```
ALTER TABLE [dbo].[Users] CHECK CONSTRAINT [FK_Users_lisanslar]
```

```
GO
```

```
ALTER TABLE [dbo].[Users] WITH CHECK ADD CONSTRAINT [FK_Users_sirketad] FOREIGN  
KEY([sirket_ID])
```

```
REFERENCES [dbo].[sirketad] ([sirket_ID])
```

```
GO
```

```
ALTER TABLE [dbo].[Users] CHECK CONSTRAINT [FK_Users_sirketad]
```

```
GO
```

Viewler

```
USE [LoginDB]
```

```
GO
```

```
/***** Object: View [dbo].[Users2] Script Date: 26.05.2023 04:14:54 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE VIEW [dbo].[Users2]
```

```
AS
```

```
SELECT          dbo.Users.id, dbo.sirketad.Sirket_AD, dbo.Users.firstLast,  
dbo.Users.email, dbo.Users.sk, dbo.Users.ls, dbo.Users.pass, dbo.Users.since
```

```
FROM            dbo.sirketad INNER JOIN
```

```
dbo.Users ON dbo.sirketad.sirket_ID = dbo.Users.sirket_ID
```

```
GO
```

Trigger



```
USE [LoginDB]
GO
```

```
/***** Object: Trigger [dbo].[update_since_trigger]    Script Date: 26.05.2023
04:16:28 *****/
SET ANSI_NULLS ON
GO
```

```
SET QUOTED_IDENTIFIER ON
GO
```

```
CREATE TRIGGER [dbo].[update_since_trigger]
ON [dbo].[Users]
AFTER INSERT
AS
BEGIN
    UPDATE users
    SET since = GETDATE()
    WHERE id IN (SELECT id FROM inserted);
END;
GO
```

```
ALTER TABLE [dbo].[Users] ENABLE TRIGGER [update_since_trigger]
GO
```

LoginDB bitışı

SQL server agent job

```
USE [msdb]
GO
```

```
/***** Object: Job [her_gun_kontrol]    Script Date: 26.05.2023 04:17:21 *****/
BEGIN TRANSACTION
DECLARE @ReturnCode INT
SELECT @ReturnCode = 0
```

```
/***** Object: JobCategory [[Uncategorized (Local)]]    Script Date: 26.05.2023
04:17:21 *****/
```

```
IF NOT EXISTS (SELECT name FROM msdb.dbo.syscategories WHERE name=N'[Uncategorized
(Local)]' AND category_class=1)
BEGIN
```

```
EXEC @ReturnCode = msdb.dbo.sp_add_category @class=N'JOB', @type=N'LOCAL',
@name=N'[Uncategorized (Local)]'
```

```
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
```

```
END
```

```
DECLARE @jobId BINARY(16)
```

```
EXEC @ReturnCode =  msdb.dbo.sp_add_job @job_name=N'her_gun_kontrol',
@enabled=1,
```

```
@notify_level_eventlog=0,
```

```
@notify_level_email=0,
```

```
@notify_level_netsend=0,
```

```
@notify_level_page=0,
```

```
@delete_level=0,
```

```
@description=N'No description available.',
```

```
@category_name=N'[Uncategorized (Local)]',
```

```
@owner_login_name=N'IT0CH\Aytac', @job_id = @jobId OUTPUT
```

```
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
```

```
/***** Object: Step [zamanlayici]    Script Date: 26.05.2023 04:17:21 *****/
```

```
EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId, @step_name=N'zamanlayici',
```

```

@step_id=1,
@cmdexec_success_code=0,
@on_success_action=1,
@on_success_step_id=0,
@on_fail_action=2,
@on_fail_step_id=0,
@retry_attempts=0,
@retry_interval=0,
@os_run_priority=0, @subsystem=N'TSQL',
@command=N'UPDATE [LoginDB].[dbo].[Users]
SET dakika_7gun = CASE WHEN DATEPART(WEEKDAY, GETDATE()) = 1 THEN 0 ELSE dakika_7gun
END,
    dakika_1gun = 0
',
@database_name=N'LoginDB',
@flags=0
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_update_job @job_id = @jobId, @start_step_id = 1
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_add_jobschedule @job_id=@jobId, @name=N'her_gun',
@enabled=1,
@freq_type=4,
@freq_interval=1,
@freq_subday_type=1,
@freq_subday_interval=0,
@freq_relative_interval=0,
@freq_recurrence_factor=0,
@active_start_date=20230522,
@active_end_date=99991231,
@active_start_time=0,
@active_end_time=235959,
@schedule_uid=N'51d03940-de24-4949-b5ec-bfbda2c5946d'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_add_jobserver @job_id = @jobId, @server_name =
N'(local)'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
COMMIT TRANSACTION
GOTO EndSave
QuitWithRollback:
    IF (@@TRANCOUNT > 0) ROLLBACK TRANSACTION
EndSave:
GO

```

Sql kısmı bitti  
Angular service kısmı

```

import { HttpClient } from '@angular/common/http';

import { Injectable } from '@angular/core';

import { Observable } from 'rxjs';

```

```

@Injectable({
  providedIn: 'root'
})

```

```

export class AuthService {

```

```
siparis1: any;
mkbilgisi:any;
// magazalogin:boolean=false;

constructor(private http: HttpClient) { }

readonly apiUrl="https://localhost:44307/api/";
apiUrl2="";
tarih:any[]=[];
//get bloğu başlangıç
getMagazalogin(): boolean {
    const magazalogin = localStorage.getItem('magazalogin');
    console.log(magazalogin);
    return magazalogin === 'true'; // localStorage'de 'true' olarak kaydedilmişse true döndürür
}

getUser( sirket_id: number): Observable<any> {
    const model = {
        sirket_id: sirket_id
    };
    return this.http.post<any>(this.apiUrl + 'User/getuser', model);
}

getSatisGunlerBirlesik():Observable<any[]>{
    return this.http.get<any>(this.apiUrl+'User/satisgunlerbirlesik');
}

getSepetGecmis():Observable<any[]>{
    return this.http.get<any>(this.apiUrl+'User/sepetgecmis');
}

getLisance():Observable<any[]>{
    return this.http.get<any>(this.apiUrl+'User/getlisance');
}
```

```
tavsiye():Observable<any[]>{
    return this.http.get<any>(this.apiUrl+'User/tavsiye');
}

getkategori():Observable<any[]>{
    return this.http.get<any>(this.apiUrl+'User/kategori');
}

getmarka():Observable<any[]>{
    return this.http.get<any>(this.apiUrl+'User/marka');
}

getveri():Observable<any[]>{
    return this.http.get<any>(this.apiUrl+'User/veri');
}

getalis():Observable<any[]>{
    return this.http.get<any>(this.apiUrl+'User/alis');
}

getsiparislerim():Observable<any[]>{
    return this.http.get<any>(this.apiUrl+'User/siparislerim');
}

hazirlanan():Observable<any[]>{
    return this.http.get<any>(this.apiUrl+'User/hazirlanan');
}

gecmis():Observable<any[]>{
    return this.http.get<any>(this.apiUrl+'User/gecmis');
}

sepet():Observable<any[]> {

    return this.http.get<any>(this.apiUrl + "User/sepet");
}

islemcisepet():Observable<any[]> {
```

```

        return this.http.get<any>(this.apiUrl + "User/islemcisepet");
    }

    getdakika():Observable<any[]> {

        return this.http.get<any>(this.apiUrl + "User/getdakika");
    }

    //get bloğu bitiş

    //insert bloğu başlangıç
    insertyeniuye(
        firstlast:string,email1:string,ls1:number, pass1:string, sirket_id: number): Observable<any> {
        const model = {
            firstlast:firstlast,email1:email1,ls1:ls1,pass1:pass1, sirket_id: sirket_id
        };

        return this.http.post<any>(this.apiUrl+"User/yeniuser", model);
    }

    insertyeniurun(urunid: number,kategoriid: number, markaid: number
        , satisfiyat: number, alisfiyat: number, urunad: string,aciklama: string ): Observable<any> {
        const model = {
            urun_id: urunid,
            kategori_id: kategoriid,
            marka_id: markaid, urun_adet: 0
            , satis: satisfiyat, maliyet: alisfiyat,
            urun_ad: urunad,aciklama: aciklama
        };

        return this.http.post<any>(this.apiUrl+"User/yeniurun", model);
    }

```

```
insertkategori(kategoriad: string): Observable<any> {  
    const model = {  
        kategori_ad: kategoriad  
    };  
  
    return this.http.post<any>(this.apiUrl+"User/insertkategori", model);  
}
```

```
insertmarka(markaad: string): Observable<any> {  
    const model = {  
        marka_ad: markaad  
    };  
  
    return this.http.post<any>(this.apiUrl+"User/insertmarka", model);  
}
```

```
insertSiparis(urunid: number, adet: number): Observable<any> {  
    const model = {  
        urunid: urunid,  
        adet: adet  
    };  
}
```

```

        return this.http.post<any>(this.apiUrl+"User/insert", model);
    }
    sepetonay(siparisid: number,siparisadet:number ,islemid:number) {
        if (islemid==1) {
            const body = { siparisid: siparisid ,durumyeni:1 ,adet:siparisadet,mk2:1};
            return this.http.put(this.apiUrl+"User/update", body);

        }else if (islemid==2) {
            const body = { siparisid: siparisid ,durumyeni:2 ,adet:siparisadet,mk2:1};
            return this.http.put(this.apiUrl+"User/update", body);
        }else if (islemid==3) {
            const body = { siparisid: siparisid ,durumyeni:3 ,adet:siparisadet,mk2:1};
            return this.http.put(this.apiUrl+"User/update", body);
        }
        else{
            const body = { siparisid: siparisid ,durumyeni:4,adet:siparisadet ,mk2:1};
            return this.http.put(this.apiUrl+"User/update", body);
        }

    }

    //insert bloğu bitiş

    //update bloğu başlangıç

    updateuser(firstlast:string,email1:string,ls1:number, email1eski: string,pass1:string, sirket_id:
number): Observable<any>
    {
        const model = {
            firstlast:firstlast,email1:email1,ls1:ls1, email1eski: email1eski,pass1:pass1, sirket_id: sirket_id
        };
    }

```

```
    return this.http.put<any>(this.apiUrl+"User/updateuser", model);  
  }
```

```
urunupdate(urunid: number,kategoriid: number, markaid: number  
  , satisfiyat: number, alisfiyat: number, urunad: string,aciklama: string ): Observable<any> {  
  const model = {  
    urun_id: urunid,  
    kategori_id: kategoriid,  
    marka_id: markaid  
    , satis: satisfiyat, maliyet: alisfiyat,  
    urun_ad: urunad,aciklama: aciklama  
  };  
}
```

```
    return this.http.put<any>(this.apiUrl+"User/updateurunler", model);  
  }
```

```
sepetsil(siparisid: number,siparisadet:number) {  
  const body = { siparisid: siparisid ,durumyeni:0,adet:siparisadet ,mk2:1};  
  return this.http.put(this.apiUrl+"User/update", body);  
}
```

```
basaribitir(siparisid: number,siparisadet:number) {  
  const body = { siparisid: siparisid ,durumyeni:3,adet:siparisadet ,mk2:1};  
  return this.http.put(this.apiUrl+"User/update", body);  
}
```

```
sepet11() {  
  const sk = localStorage.getItem('sk');  
  const ls = localStorage.getItem('ls');  
  const mk = localStorage.getItem('mk');  
  const body = { mk2: this.mkbilgisi };  
  console.log(localStorage.getItem('mk'));
```



```

        return this.http.post(this.APIUrl+"User/sepet", body);
    }

    urunsil(urunid: number): Observable<any> {
        const model = {
            urun_id: urunid,

        };

        return this.http.put<any>(this.APIUrl+"User/urunsil", model);
    }

    adetekle(urunid: number,urunadet: number): Observable<any> {
        const model = {
            urun_id: urunid,
            urun_adet: urunadet,

        };

        return this.http.put<any>(this.APIUrl+"User/adetekleurunler", model);
    }

    dakikaekle(dakika: number): Observable<any> {
        const model = {
            id2: localStorage.getItem('id'),
            dakika: dakika
        };

        return this.http.put<any>(this.APIUrl+"User/dakikaekle", model);
    }

    updateAPIUrl(sk: string) {
        // this.APIUrl2 = `${sk}`;
    }

```

```
}
```

```
//update bloğu bitiş
```

```
//delete bloğu başlangıç
```

```
usersil(id: number): Observable<any> {
```

```
    const model = {
```

```
        id1: id,
```

```
    };
```

```
    return this.http.put<any>(this.apiUrl+"User/deleteuser", model);
```

```
}
```

```
//delete bloğu bitiş
```

```
    setMagazallogin(value: boolean): void {
```

```
        localStorage.setItem('magazallogin', String(value));
```

```
}
```

```
loginruser(logininfo:Array<String>){
```

```
    return this.http.post(this.apiUrl+"User/LoginUser",{
```

```
        email:logininfo[0],
```

```
        pass:logininfo[1],},
```

```
        {responseType:'text'},
```

```
    );
```

```
}
```

```
logout(): void {
```

```
    this.apiUrl2="";
```

```
    localStorage.clear();
```

```
    localStorage.removeItem('sk');
```

```
    localStorage.removeItem('ls');
```

```
    localStorage.removeItem('mk');
```

```
}}
```

## **PART I Bitiş**



**DOKUZ EYLÜL ÜNİVERSİTESİ  
İZMİR MESLEK YÜKSEKOKULU  
BİLGİSAYAR TEKNOLOJİLERİ BÖLÜMÜ**

# **DMA DATA MANAGEMENT**

# APPLICATION

Umut G ng r 2021730029

 zmir

Mayıs 2023

Tez Danıřmanı

 gr. G r. İsmail D řMEZ

## PART II

### İ indekiler

Onay .....	3
Etik İlkelere Uygunluk Beyanı.....	4
Teřekk�r .....	5
1.proje fikrinin bilgisinin bulunması.....	<b>Hata! Yer iřareti tanımlanmamıř.</b>
2.arařtırma s�reci tasarlanması (05.03.2023) .....	7
3.zaman planlaması .....	7
4.toplantı kararları .....	10
5.gereksinim analizi .....	13
6.tasarım .....	<b>Hata! Yer iřareti tanımlanmamıř.</b>
7.kodlama .....	<b>Hata! Yer iřareti tanımlanmamıř.</b>
1.Tasarım i�in k�t�phane arayıřı .....	118
2.Programların Kurulması .....	122
3.Tasarım Fikirlerinin Tartıřılması .....	123
4. K�t�phaneleri �ğrenme .....	124
5. HTML Sayfalarının Tasarlanması .....	125

6.HTML kodları.....	140
7.Yönlendirmeler ve Etkileşimler (TS kodları).....	195
8.CSS Kodları .....	303
9.Kod Açıklamaları .....	306
10.test ve hataların bulunması. ....	308
11.proje paketleme ve teslim. ....	309

AuthService adındaki bu service Angular uygulamasında kullanıcı kimlik doğrulama, veri alışverişi ve yönetimi gibi işlevleri gerçekleştiren bir servis sağlar. Aşağıda kodun yapılandırması ve içerdiği özellikler açıklanmaktadır:

- 'HttpClient' ve 'Injectable' gibi Angular'ın yerleşik modüllerini kullanarak servisin gerekli bağımlılıkları enjekte etmesi sağlanmıştır.
- 'APIUrl' değişkeni, API'nin kök URL'sini içermektedir. Bu değişken, API ile iletişim kurmak için kullanılacaktır.
- 'getMagazalogin()' fonksiyonu, 'localStorage' üzerinde kaydedilmiş olan 'magazalogin' değerini döndürerek kullanıcının giriş yapmış olup olmadığını kontrol eder.
- 'getUser()' fonksiyonu, belirli bir şirket ID'sine sahip kullanıcıyı getirmek için API'yi çağıran bir HTTP POST isteği yapar.
- Kodun geri kalanında, çeşitli API çağrıları için HTTP GET ve POST istekleri tanımlanmıştır. Örneğin 'getsatisgunlerbirlesik()', 'getsepetgecmis()', 'getlisance()' gibi

fonksiyonlar, ilgili API rotalarını çağırmak için `HttpClient` kullanır ve sonuçları bir `Observable` nesnesi olarak döndürür.

- Ayrıca, `insertyeniuye()`, `insertyeniurun()`, `insertkategori()`, `insertmarka()` gibi fonksiyonlar da yeni veri eklemek için HTTP POST istekleri yapar.

- `updateuser()`, `urunupdate()`, `sepetsil()`, `basaribitir()` gibi fonksiyonlar mevcut veriyi güncellemek için HTTP PUT istekleri yapar.

- `usersil()` fonksiyonu ise bir kullanıcıyı silmek için HTTP PUT isteği yapar.

- `loginruser()` fonksiyonu, kullanıcı girişi için API'yi çağıran bir HTTP POST isteği yapar ve kullanıcıyı doğrular.

- `logout()` fonksiyonu, kullanıcının oturumunu sonlandırır ve ilgili verileri temizler.

## Tasarım Aşamaları

### 1.Tasarım için kütüphane arayışı

Proje tasarımı için kütüphane arayışı yaparken ilk olarak kendimin tecrübeli olduğu bir dil olan **javascript** çatısı altında çalışan ve kullanılabilir olan kütüphaneleri aradım. Sonuçlar arasında **Angular** adlı bir kütüphane gördüm ve projede kullanımı avantajlı bir sonuç vereceğini düşünerek **Angular** kütüphanesini kullanma kararı aldım. Diğer kütüphaneler temel olarak nesne ve nesne özelliklerinin hazır olarak kullanmamı kolaylaştırmak ya da kodların düzenli ve birden fazla yere kolay entegrasyonunu sağlamak için kullanılan ekstra kütüphaneler olarak kullanılmıştır. Sonuç olarak proje tasarımı için **Angular**, **SCSS**, **JSCHART**, **BOOTSTRAP** gibi kütüphaneler kullanılmıştır. Açıklaması, Projede kullanım amaçları, neden tercih edildiği ve faydaları aşağıda ayrıntılı bir şekilde verilmiştir.

### 1.1 Angular

#### Açıklama

Angular, web uygulamalarını geliştirmek için kullanılan bir JavaScript framework'üdür. Güçlü bir yapıya sahiptir ve büyük ölçekli projeler için idealdir. Angular, bileşen tabanlı bir yapı sunar, böylece kodunuzun yeniden kullanılabilirliği artar ve projenin daha sürdürülebilir olmasını sağlar.

## Kullanım Amaçları

Angular kütüphanesi bu projede kullanım amaçları, modüler yapısı sayesinde tasarımı kolaylaştırması, bileşen adlı sisteme sahip olması ve bileşenleri oluşturduktan sonra birden fazla sayfada kullanabilme avantajı, tekrarlı kodu engelleme, etkileşimlerin bulunduğu kod sayfasının 'ts' uzantılı dosyalarda yazılması ve ts dosyalarının kodların düzenlenmesini yada hata yapılmasını engelleme özellikleri , javascript çatısının altında geliştirilmiş olması, HTTP ile entegresyonun kolay sağlanabilmesi , servis modülünün kullanım avantajları , API ile uyumu, kendine has eventler bulunması ve bu eventlerin projede sık sık kullanılması.

## Tercih Nedenleri Ve Faydaları

- **Modüler yapı:** Angular, bileşen tabanlı bir yapı sunar. Bu, projenin farklı bileşenlerine (stok takibi, raporlama, güncelleme vb.) bölünerek daha modüler bir kod yapısı oluşturmanızı sağlar. Bu şekilde, her bileşenin bağımsız olarak yönetilmesi ve yeniden kullanılabilir olması mümkün olur.
- **Verimli geliştirme:** Angular, geliştirme sürecini hızlandırır ve verimli hale getirir. Özellikle büyük ölçekli projelerde, Angular'ın sağladığı yapı ve özellikler sayesinde kodun daha organize olması, test edilebilirlik, hata yönetimi ve kodun yeniden kullanılabilirliği gibi avantajlar elde edebilirsiniz. Bu da projenin geliştirme sürecini daha verimli hale getirir.
- **İyi bir topluluk desteği:** Angular, geniş bir kullanıcı topluluğuna sahiptir ve bu topluluk sürekli olarak yeni kaynaklar, kılavuzlar ve yardımcı materyaller oluşturur. Bu, projenizin geliştirilmesi ve karşılaşılan sorunların çözülmesi sürecinde size destek sağlar. Ayrıca, topluluk destekli framework'ler genellikle güncellemelerle ve hata düzeltmeleriyle daha iyi desteklenir.
- **İyi bir veri bağlama (data binding) mekanizması:** Angular, iki yönlü veri bağlama (two-way data binding) özelliği sunar. Bu özellik, stok bilgilerinin gerçek zamanlı olarak güncellenmesini sağlar. Böylece, kullanıcı bir stok girişi yaptığında veya stokta bir değişiklik olduğunda, bu değişiklikler anında ekrana yansır ve kullanıcıya geri bildirim verilir.
- **Harici kütüphanelerin entegrasyonu:** Angular, harici kütüphanelerin kolayca entegre edilebilmesini sağlar. Örneğin, grafik veya tablo kütüphaneleri gibi ek işlevselliği projenize ekleyebilirsiniz. Bu da stok kontrol sisteminizin daha zengin ve kullanışlı hale gelmesini sağlar.

## 1.2 SCSS

### Açıklama

SCSS, CSS preprocessor'lerinden biridir ve CSS yazımını daha modüler, daha düzenli ve daha kolay hale getirir. Stil sayfalarınızı daha iyi organize edebilir, tekrar kullanılabilirlik sağlayabilir ve stilinizi daha etkili bir şekilde yönetir.

## Kullanım Amaçları

SCSS kütüphanesinin kullanılma amaçları, CSS yazımını daha modüler ve düzenli hale getirerek geliştirme sürecini kolaylaştırması, sayfaları daha iyi organize edebilir ve stil kurallarını daha etkili bir şekilde yönetilmesini kolaylaştırır, değişkenler kullanarak tekrar eden değerleri tek bir yerden yönetilebilmesi ve kolayca güncellenmesi, fonksiyonlar ise CSS kodunuzda tekrar eden mantıksal işlemleri gerçekleştirmenizi sağlar, böylece kodun daha temiz ve daha az tekrar eden yapıya sahip olmasını sağlar.

### Tercih Nedenleri Ve Faydaları

- **Modülerlik:** SCSS, CSS kodunuzu modüller halinde yazmaya olanak tanır. Bu şekilde, stil sayfalarını bileşenlere veya özelliklere göre ayrı ayrı düzenleyebilir ve yönetebilir olmasını sağlar. Bu modüler yapı, kodun yeniden kullanılabilirliğini artırır ve bakım sürecini kolaylaştırır.
- **Değişkenler:** SCSS, değişkenleri destekler. Değişkenler sayesinde, belirli renkler, boyutlar veya stiller gibi tekrar eden değerleri tek bir yerden yönetebilir yapar. Böylece, değerleri tek bir yerde güncellenmez durumunda tüm stil sayfalarında otomatik olarak güncellenir.
- **Nesting:** SCSS, stil kurallarını iç içe geçirme (nesting) özelliğine sahiptir. Bu, stil kurallarını daha anlaşılır ve düzenli bir şekilde gruplayabilmeyi sağlar.
- **İmport ve Mixin'ler:** SCSS, import özelliği sayesinde stil sayfalarınızı bölümlere ayırabilir ve daha iyi bir yapılandırma sağlayabilir. Ayrıca mixin'lerle stil sayfalarınızda tekrar eden stilleri daha kolay bir şekilde tanımlayabilir ve kullanabilir olmasını sağlar. Bu, kodunuzun daha temiz ve daha az tekrar eden yapıya sahip olmasını sağlar.

## 1.3 JSCHART

### Açıklama

JSChart, bir JavaScript tabanlı bir grafik kütüphanesidir. Bu kütüphane, web uygulamalarında grafik ve veri görselleştirmeleri oluşturmak için kullanılır. JSChart, kullanıcıların verilerini çeşitli grafik türlerinde (çizgi grafikleri, sütun grafikleri, pasta grafikleri, dağılım grafikleri vb.) görselleştirmelerini sağlar.

### Kullanım Amaçları

JSCHART kütüphanesinin kullanılma amaçları, grafik şemalarını oluşturması, veri gruplarından gelen veriler ile entegrasyonu olması ve otomatik olarak veriye göre kendini güncellemesi, kütüphanenin **js** ile uyumlu olması, şemaların farklı şekilde kullanıcıya sunulabilmesi, şemaların düzenli ve temiz bir görüntü sunması, responsive bir tasarım sunması, kullanımı kolay ve anlaşılabilir olması

### Tercih Nedenleri Ve Faydaları



- **Çeşitli grafik türleri:** JSChart, kullanıcıların farklı grafik türlerini kullanarak verilerini görselleştirmelerini sağlar. Örneğin, zaman serisi verilerini çizgi grafikleriyle, kategorik verileri sütun grafikleriyle veya oranları pasta grafikleriyle gösterebilir
- **Özelleştirilebilirlik:** JSChart, grafiklerin görünümünü ve davranışını özelleştirmeye olanak tanır. Renkler, etiketler, eksenler, arka planlar gibi grafik bileşenlerini özelleştirebilir ve kullanıcı deneyimini iyileştirilebilmesini sağlar.
- **Veri bağlama (data binding):** JSChart, verileri dinamik olarak grafiklere bağlamanıza olanak tanır. Bu sayede, veri setlerindeki değişiklikler otomatik olarak grafiklere yansıtılır ve güncellenir. Bu özellik, gerçek zamanlı veri güncellemeleri veya dinamik veri görselleştirmeleri için oldukça yararlıdır.
- **Responsif tasarım:** JSChart, responsif tasarım özelliklerine sahiptir. Bu, grafiklerin farklı ekran boyutlarına ve cihazlara uyumlu olmasını sağlar. Kullanıcılar, grafikleri mobil cihazlarda veya farklı ekran çözünürlüklerinde sorunsuz bir şekilde görüntüleyebilir.
- **Animasyonlar:** JSChart, grafiklere animasyonlar ekleyerek kullanıcı deneyimini iyileştirebilir. Grafiklerin yüklenme veya veri güncellenme süreçlerinde akıcı geçişler sağlayarak kullanıcıların verileri daha etkileyici bir şekilde görmelerini sağlar.

## 1.4 Bootstrap

### Açıklama

Bootstrap, web uygulamaları ve siteleri geliştirmek için popüler bir CSS ve JavaScript kütüphanesidir. Twitter tarafından oluşturulan ve açık kaynaklı olarak yayınlanan Bootstrap, kullanıcı arayüzünü oluşturmak ve düzenlemek için bir dizi önceden tasarlanmış bileşen, stil ve etkileşim sunar.

### Kullanım Amaçları

Bootstrap kütüphanesinin kullanılma amaçları, hazır şemalarla ve sistemleri ile tasarımın kolayca tasarlanmasını sağlması, düzenli şık bir görüntüye sahip olması, sürekli olarak güncel kalması, responsive bir yapıya sahip olması, kullanıcının alışkın olduğu bir yapıya sahip olması, grid,row,column,flex gibi sistemler bulundurması ve bu sistemleri hazır bir şekilde sunması.

### Tercih Nedenleri Ve Faydaları

- **Duyarlı (Responsive) Tasarım:** Bootstrap, duyarlı tasarım prensiplerine uygun olarak geliştirilmiştir. Bu sayede, web uygulamanız veya siteniz farklı ekran boyutlarına ve cihazlara otomatik olarak uyum sağlar. Kullanıcılar, masaüstü bilgisayarlardan mobil cihazlara kadar farklı platformlarda sorunsuz bir deneyim yaşar.
- **Önceden Tasarlanmış Bileşenler:** Bootstrap, kullanımı kolay ve özelleştirilebilir bir dizi önceden tasarlanmış bileşen sunar. Butonlar, navigasyon çubukları, form alanları,

kartlar, modallar, slaytlar ve daha birçok bileşen, geliştiricilere hızlı ve konsistent bir şekilde kullanılabilir. Bu bileşenler, zaman kazandırır ve kullanıcı arayüzünü tutarlı bir şekilde oluşturmanıza yardımcı olur.

- **Grid Sistemi:** Bootstrap, bir grid sistemi üzerine kuruludur. Grid sistemi, web sayfalarını bölerek içeriklerin düzenli bir şekilde yerleştirilmesini sağlar. Bu, sayfanızın yapısını oluştururken kolaylık sağlar ve responsive tasarım için önemli bir temel unsurdur.
- **Stil Şablonları ve Temalar:** Bootstrap, farklı stil şablonları ve temalar sunar. Bu şablonlar ve temalar, web uygulamanıza veya sitesine hızlı bir şekilde görsel olarak çekici bir görünüm kazandırmanıza olanak tanır. İhtiyaçlarınıza ve marka kimliğinize uygun olarak temaları özelleştirebilirsiniz.
- **JavaScript Bileşenleri:** Bootstrap, kullanıcı etkileşimlerini geliştirmek için bir dizi JavaScript bileşeni içerir. Carousel (slayt gösterisi), dropdown menüler, modal pencereler, tooltips (ipuçları) ve daha birçok etkileşimli öğe kullanıcı deneyimini zenginleştirir.

## Özet

Kısacası her kütüphanenin kendine has görevleri ve kullanım amaçları vardır, bu kütüphaneler pozitif yönlerinden dolayı projeye uygun görülmüş ve şu an ki tecrübeme dayanarak kendi bilgimi en iyi şekilde entegre edebileceğimi düşündüğüm için seçilmiştir.

## 2. Programların Kurulması

### 2.1 Tasarım için kullanılan ana program

#### 2.1.1 Visual Studio Code

Microsoft tarafından geliştirilen ve açık kaynaklı bir metin düzenleyicisi ve entegre geliştirme ortamıdır. Birçok platformda kullanılabilen VS Code, geliştiricilere çeşitli programlama dilleriyle çalışırken verimli bir şekilde kod yazma ve projelerini yönetme imkanı sunar.

### 2.2 Eklentiler

- **Angular Language Service**

Angular projelerinde geliştirme yaparken geliştiricilere yardımcı olan bir araçtır. Bu eklenti, Angular uygulamalarının kodunu analiz eder ve hızlı bir şekilde geri bildirim sağlar, böylece daha hızlı ve hatasız bir şekilde kod yazılmasını sağlar.

- **Bootstrap 5 Quick Snippets**

Bu eklenti, Bootstrap 5'e özgü kod parçacıklarını içerir ve geliştiricilere hızlı bir şekilde Bootstrap bileşenlerini oluşturma imkanı sağlar.

- **Live Sass Compiler**

Bu eklenti, SASS (Syntactically Awesome Style Sheets) veya SCSS (Sassy CSS) dosyalarını otomatik olarak CSS'ye derleyerek canlı bir şekilde güncellenmesini sağlar.

- **SCSS Formatter**

Bu eklenti, SCSS (Sassy CSS) dosyalarını otomatik olarak biçimlendirir ve düzenler, böylece kodun okunabilirliğini artırır ve tutarlı bir stile sahip olmasını sağlar.

## **Paket Yönetici**

### **NPM(Node Package Manager)**

JavaScript tabanlı uygulamaların paket yönetimi için kullanılan bir paket yöneticisidir. NPM, Node.js ile birlikte gelir ve JavaScript projelerinde harici bağımlılıkları yönetmek, paketleri indirmek, yükseltmek, kaldırmak ve projede kullanmak için kullanılır.

### **İndirilen paketler**

- **Bootstrap**
- **JSCHART**
- **Angular**
- **SCSS**

## **Özet**

Kullanılan program, eklentiler, paket yöneticisi, paketler belirtilmiş ve açıklanmıştır.

## **3.Tasarım Fikirlerinin Tartışılması**

### **Tartışma Sonucu**

Renk paletinin açık ve canlı renklerde olması gerektiği ve modern şık bir dizayn yapılması gerektiği kararı aldık ve renk paletini aşağıdaki gibi belirledik.

### **Renk Paleti :**

```
$primary:
  #1d4b79
;
$secondary:
  #ff8b07
;
$warning:
  #1ABC9C
;
$info:
  #03A9F4
;
$success:
  #FFC107
```

## 4. Kütüphaneleri Öğrenme

Proje için kütüphaneler hakkında bilgi almam ve öğrenmem gereken çok şey vardı. Bu yüzden bu zaman aralığında aşağıdaki gibi bir öğrenim planı kullandım.

### 4.1 Bootstrap

- Temel nesneleri öğrenme
- Temel nesnelerde özelleştirme
- Genel özelleştirmeleri öğrenme
- Nesne hizalama, pozisyon yönetimi, sistem yönetimi
- Nesne etkileşimleri
- Tüm bilgileri entegre ederek kullanabilme

### 4.2 Angular

- Sistem hakkında bilgilenme işleyiş hakkında bilgilenme
- Bileşen ve ya modül sistemi hakkında bilgi edinme
- TypeScript hakkında bilgi edinme
- HTTP ve Servis hakkında bilgi edinme
- Router ile sayfa yönlendirme
- Javascript bilgimi pekiştirme ve TypeScript'e uyarlama

- Etkileşim kullanımını öğrenme ve sayfalar arası geçiş yapma
- HTTP metodlarından gelen bilgileri sayfaya yansıtma ve filtreleme

### **4.3 JSCHART**

- Sistem hakkında bilgi edinme
- Angular ile kullanımı
- Bileşende kullanma
- Grafiklerin veri kümesine veri eklenmesi ve grafik görünüşünün, tipinin özelleştirilmesi
- Güncel olarak veri aktarımı

### **4.4 SCSS**

- CSS ile SCSS arasındaki farkları öğrenme
- Class tanımlamaları ve iç içe kullanımı
- Sürekli sayfaya uygulaması
- Angular'a entegresi
- Mixin ve İmport kullanımını öğrenme
- CSS özelliklerini pekiştirme ve SCSS'e uyarlama

## **Özet**

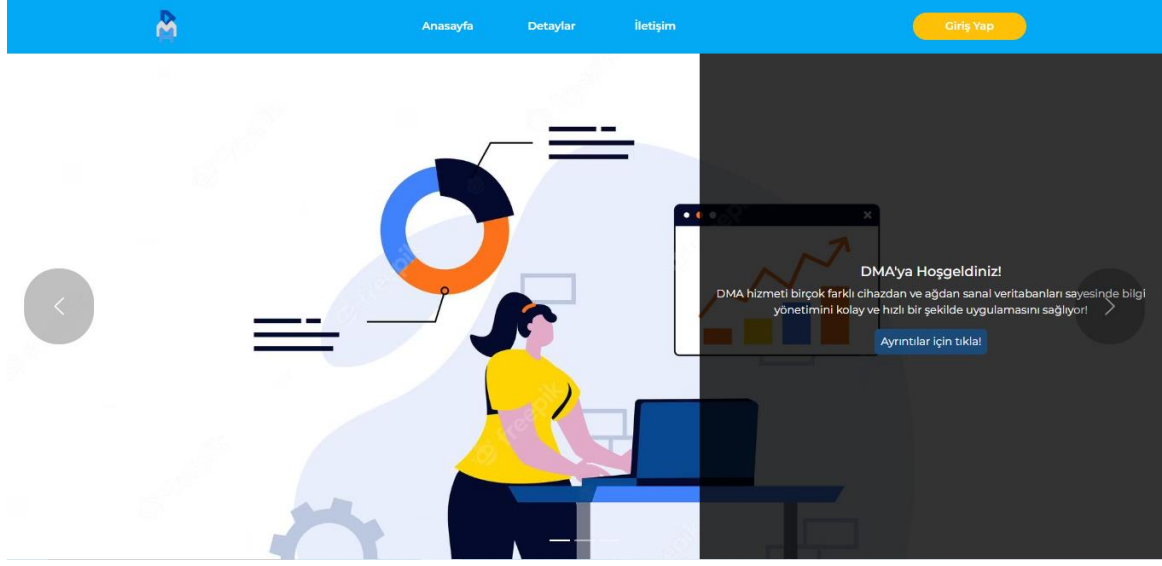
Proje tasarımı yaparken öğrenmem gereken konular ve içerikler belirtilmiş ve sırasıyla öğrenim planım yazılmıştır.

## **5. HTML Sayfalarının Tasarlanması**

Html sayfalarını tasarlarken angular ile uyumlu şekilde çalışacak şekilde tasarladım genel olarak ferah şık ve sade bir dizayn yapmayı hedefledim. Kimler için yazıldığı yanında belirtilerek başlık olarak yazılmıştır.

### **5.1. Anasayfa - Kullanıcılar**

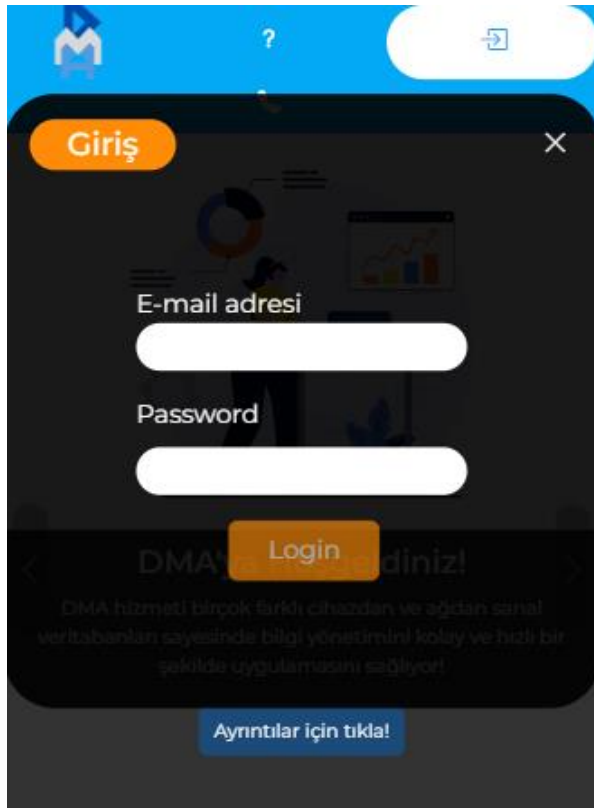
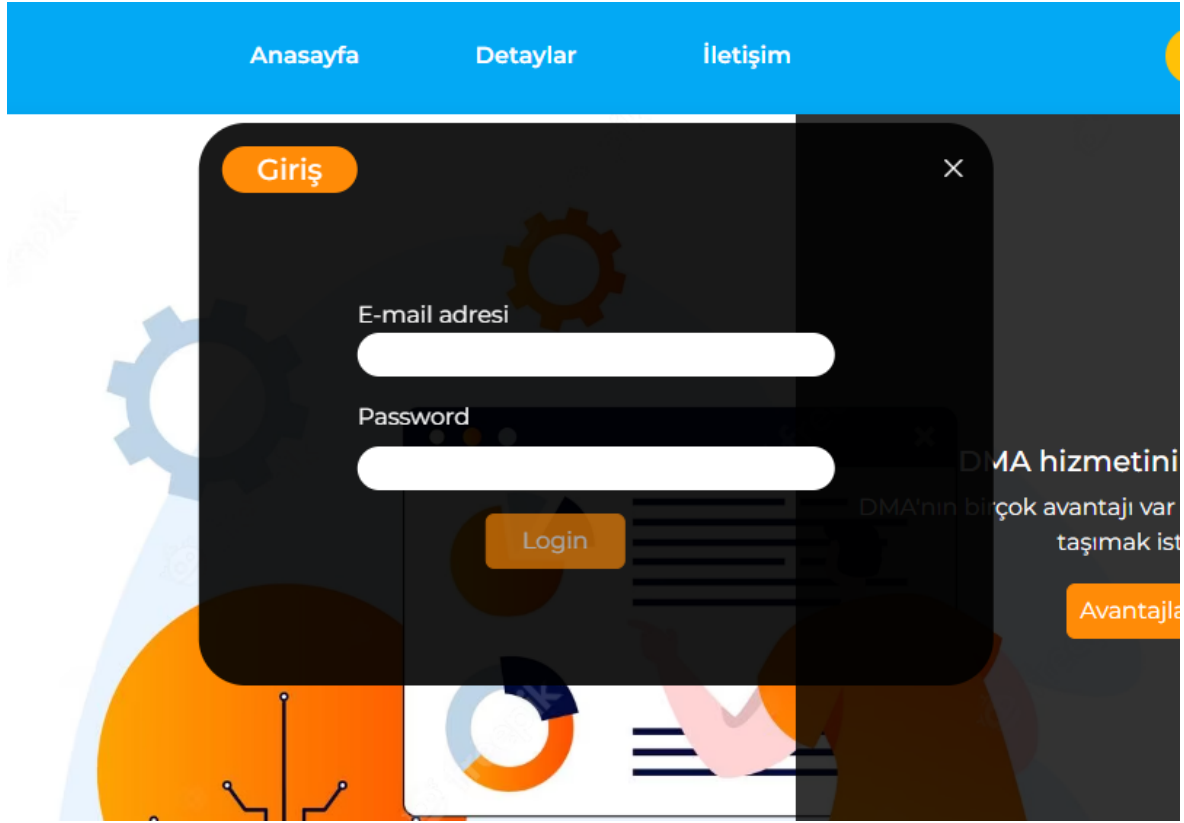
## - Giriş



### Açıklama

Yukarıdaki ve soldaki görsel kullanıcının ilk kez karşılaştığı görselleri temsil etmektedir soldaki görsel responsive olarak sayfa görünümü değiştiğinde bu hali aldığını göstermek için konulmuştur. Menüdeki yazılar direkt sayfa ya da bölümlere yönlendirir. Giriş yap butonu ise login ekranının açılmasını ve sisteme giriş yapılabilmesini sağlar. navbar navbar.html den alınmıştır. Carousel ise **Content.component.html** den alınmıştır.

## - Giriş ekranı



### Açıklama

Giriş ekranı sisteme giriş yapan hesap bilgilerini alıp sisteme girişi onaylayan ya da reddeden kısımdır. Çeşitli kontrollerden geçirilerek giriş yapılmasına izin verilir. Aynı şekilde responsive görüntüsü verilmiştir. Login ekranı **navbar.component.html** den alınmıştır.

### - İçerik



## DMA Nedir?

DMA bir ticari amaç güden işletmeler için verilen bir hizmettir. DMA hizmeti, iş ile alakalı olan verilerinizi sanal bir veritabanında tutar ve güncel olarak bilgileri işler bu sanal veritabanında isteğiniz şekilde yönetmenizi ve akıllı sistemi ile işletmenizi bir adım ileri taşımayı amaçlar.

### Açıklama

Kısa bir tanıtım paragrafı ile sistemi anlattık.

### - Avantajlar ve Footer

#### DMA'nın Avantajları!




**Sanal Veritabanı**

Sanal veritabanları ile internetiniz olduğu her anda ulaşılabilirlik sağlıyor ve size daha kısa sürelerde isteklerinizi gerçekleştirebiliyoruz.



**Erişebilirlik**

DMA'da erişebilirlik tam seviyede! İstedığınız şekilde veritabanınızı düzenleyebilir veya altınızda çalışan işçileri yetkilendirebilirsiniz!



**Akıllı Sistem**


DMA hizmeti istatiki veriler üzerinden size çeşitli tavsiyeler verir ve bir nevi sizin küçük bir yardımcınız olur.



**İstatiki Veri Analizi**

DMA hizmeti Veritabanındaki verileri çeşitli işlemlerden geçirerek size kolay anlaşılabilir grafik ve tablolarla size sunar.

Logo



**İletişim Bilgileri**

E-posta

[dma.info.tr@gmail.com](mailto:dma.info.tr@gmail.com)

Tel

+90 534 035 2964

+90 545 287 57 91

**Hakkımızda!**

DMA projesini 2 kişi geliştirmiştir bu iki kişi şuanda Dokuz Eylül Bil. Prog. bölümünde aktif olarak okumaktadır. Tasarım kısmı Umud Güngör, Arka İşlem kısmını Aytaç Melih Kılıç üstlenmiştir. DMA projesi Okul projesi olarak sunulmak için hazırlanmıştır.





## Açıklama

Bu bölümlerde sistemin avantajları ve iletişim bilgileri verilmiştir. Sistemin avantajları toplam 4 tane ana avantajı verilmiş ve bu verilere göre dizayn edilmiştir. İletişim bilgileri ise sistem hakkında bir bilgi alma yada kayıt olma işlemi için verilmiştir. Aynı şekilde responsive dizayn fotoğrafları gösterilmiştir. **Content.component.html** den alınmıştır.

## 5.2 Yönetici Ekranları - Yönetici

### 5.2.1 Grafik ve Analiz Ekranı

#### - Tavsiye ekranı ve ekran menüsü



## Açıklama

Yönetici ekranı 4 alandan oluşmaktadır. Grafik ve analiz, Raporlar, Sipariş Geçmişi ve Hesaplar. Grafik ve analiz ekranında, yönetici ürünlerin belirli başlıklar altında gelen verileri grafikte ve analiz kısmında görüntüleyebilir ve çeşitli şekilde filtreleyebilir. Raporlar ekranında verilerin toplanıp derlenmesi sonucu çeşitli sonuçları yöneticiye gösterir. **Sipariş geçmişi** firmasının aldığı ve işlemi biten siparişlerin gösterildiği yerdir ve sadece yönetici erişmez **birden fazla yerde kullanıldı** bu yüzden **bir daha açıklamayacağım**. Hesaplar ekranı, yönetici şirkette bulunan işçilere hesap yaratabilir düzenleyebilir veya silebilir. Tavsiye ekranı ise yöneticiye ürünlerin belli zaman aralıklarındaki başlıklar altında bilgiler ve öneriler verir.

## - Grafik ve Analiz





### Açıklama

Grafik alanı sağ köşedeki select ten seçilen başlığa, sol üstteki grafik tipine ve zaman aralığına ve son olarak marka ve kategori olarak filtrelenebilir hazırlanmıştır. **Zaman aralığı butonları** (Özel tarih hariç) Grafik ve Analiz sekmesindeki **bütün ekranları etkilemektedir**. Analiz kısmı, filtrelere göre ilk 10 ve daha fazla sonuç varsa hepsini gösterir. Solda responsive tasarımı gösterilmiştir. **yonetici.component.html** 'den alındı.

## 5.2.2 Raporlar

### - Rapor Ekranları



## Açıklama

Yönetici burada çeşitli zaman aralıklarına göre stok bilgilerini, satış bilgilerini, performans bilgilerini ve finans bilgilerini görüntüleyebilir. Stok bilgilerinde stok takibi yapabilir, Satış bilgilerinde belirli satış bilgilerine ulaşabilir, Performans bilgileri, sipariş durum ortalaması ve sipariş tamamlama ortalamasına ulaşılabilir ve finans bilgilerinde ise toplam maliyet ve satış fiyatlarını görüntüleyebilir ve kar oranını hesaplar.**yonet-rapor.component.html** dan alındı.

### 5.2.3 Sipariş Geçmişi

Ürün id	Ürün ad	Kategori	Marka	Adet	Firma	Tutar	Sipariş durum	Sipariş tarihi	Teslim tarihi
1	koltuk takımı	ev eşyası	doğtaş	16	AH firması	320000 TL	sipariş gönderildi	24/05/2023	24/05/2023
3	derin dondurucu	ev eşyası	arçelik	5	AH firması	90000 TL	sipariş iptal	23/05/2023	24/05/2023
2	pense	ev eşyası	bosh	17	AH firması	935 TL	sipariş gönderildi	24/05/2023	24/05/2023
2	pense	ev eşyası	bosh	3	AH firması	45 TL	sipariş gönderildi	12/05/2023	18/05/2023
18	kesme tahtası	mutfak aletleri	yeşil aksesuar	1	AH firması	50 TL	sipariş gönderildi	12/05/2023	18/05/2023
1	koltuk takımı	ev eşyası	doğtaş	31	AH firması	93 TL	sipariş gönderildi	18/05/2023	18/05/2023
1	koltuk takımı	ev eşyası	doğtaş	1	AH firması	3 TL	sipariş gönderildi	12/05/2023	16/05/2023
5	scs	ürün kalktı	arçelik	17	AH firması	1530 TL	sipariş gönderildi	11/05/2023	15/05/2023

Firma	Tutar	Sipariş durum
AH firması	320000 TL	sipariş gönderildi
AH firması	90000 TL	sipariş iptal
AH firması	935 TL	sipariş gönderildi
AH firması	45 TL	sipariş gönderildi
AH firması	50 TL	sipariş gönderildi
AH firması	93 TL	sipariş gönderildi
AH firması	3 TL	sipariş gönderildi

### Açıklama

Yönetici burada işlemi biten siparişleri ayrıntılı şekilde görüntüleyebilir.

**Bütün tablolar bu şekilde responsive olarak tasarlanmıştır aşağıda da aynı şekilde bir scrool vardır. gecmis.component.html den alınmıştır.**

### 5.2.4 Hesaplar

Hesaplar					Personel Ekle
Personel E-mail	Personel Şifre	Personel Rol	Personel İsim	İşlem	
umutgungormez@gmail.com	123456	yönetici	umut	Düzenle Hesap Hareketleri	
aytacmelihkili@gmail.com	123456	yönetici	aytac	Düzenle Hesap Hareketleri	
depol@gmail.com	123456	depo	depol	Düzenle Sil Hesap Hareketleri	
islemci@gmail.com	123456	işlemci	islemci	Düzenle Sil Hesap Hareketleri	

Tüm Hesap Hareketleri

Toplam Hesap Sayısı

5

Personel İsim	Personel Hesap aktifliği (günlük)	Personel Hesap aktifliği (haftalık)
umut	0 dk	0 dk
aytac	3 dk	10 dk
depol	0 dk	5 dk
magaza1	0 dk	18 dk



### Hesaplar

Personel Ekle

Personel Rol	Personel İsim	İşlem
yönetici	umut	<button>Düzenle</button> <button>Hesap Hareketleri</button>
yönetici	aytac	<button>Düzenle</button> <button>Hesap Hareketleri</button>
depo	depo1	<button>Düzenle</button> <button>Sil</button> <button>Hesap Hareketleri</button>
işlemci	islemci1	<button>Düzenle</button> <button>Sil</button> <button>Hesap Hareketleri</button>

### Tüm Hesap Hareketleri

## Toplam Hesap Sayısı

5

Personel İsim	Personel Hesap aktifliği (günlük)	Personel Hesap aktifliği (haftalık)
umut	0 dk	0 dk
aytac	3 dk	10 dk
depo1	0 dk	5 dk
magaza1	0 dk	18 dk

### Açıklama

Hesaplar kısmında yönetici kendi veya başka bir hesabı düzenleyebilir ve kendi hesabı hariç hesapları silebilir. Hesap hareketleri ekranı ise hesapta ne kadar aktif kaldıklarını haftalık ve günlük olarak gösterir. Responsive görüntüleri verilmiştir. **yonet-hesap.component.html** den alınmıştır.

## 5.3 İşlemci Ekranı - İşlemci

### 5.3.1 İstekler

#### - İstekler ve İşlemci alanları

İstekler		ürünler ve ayrıntıları			hazırlanan ürünler		Sipariş Geçmişi			
Ürün Id	Ürün ad	Kategori	Marka	stok	onaylanabilir adet	sipariş adeti	Fiyat	firma	Tarih	İşlem
1	koltuk takımı	ev eşyası	doğtaş	183	173	2	20000	AH firması	15/05/2023	<a href="#">onayla</a> <a href="#">iptal et</a>
1	koltuk takımı	ev eşyası	doğtaş	183	173	2	20000	AH firması	15/05/2023	<a href="#">onayla</a> <a href="#">iptal et</a>
1	koltuk takımı	ev eşyası	doğtaş	183	173	1	20000	AH firması	15/05/2023	<a href="#">onayla</a> <a href="#">iptal et</a>
3	derin dondurucu	ev eşyası	arçelik	269	251	10	18000	AH firması	19/05/2023	<a href="#">onayla</a> <a href="#">iptal et</a>
4	biblo	aksesuar	yeşil aksesuar	118	118	3	45	AH firması	19/05/2023	<a href="#">onayla</a> <a href="#">iptal et</a>
3	derin dondurucu	ev eşyası	arçelik	269	251	3	18000	AH firması	19/05/2023	<a href="#">onayla</a> <a href="#">iptal et</a>
15	kağıt	kırtansiye/	ve-ge	58	58	22	25	AH firması	19/05/2023	<a href="#">onayla</a> <a href="#">iptal et</a>

#### Açıklama

İşlemci bu sayfada gelen siparişlere bakıp onaylayıp ya da reddedip işleme sokuyor. İşlemci alanları ise istekler, ürünler ve ayrıntıları, hazırlanan ürünler ve sipariş geçmişi olarak düzenlenmektedir. Ürünler ve ayrıntılar, ürünlerin hakkında belli bilgiler veriliyor ve düzenleme veya yaratma işlemleri yapılıyor aynı zamanda marka veya kategori de eklenebiliyor.

Hazırlanan siparişler de ise takvim ile günlerdeki sipariş durumlarını ve olup olmadığını anlayabiliyorsunuz ve sipariş iptalinde bulunabiliyorsunuz. Geçmiş siparişte üstte bahsedildi.

**Islemci.component.html den alındı.**

### 5.3.2 Ürünler ve ayrıntıları

Ürün Id	Ürün ad	Kategori	Marka	Ayrıntılar	Stok	satılabilir stok	Fiyat	Maliyet	
1	koltuk takımı	ev eşyası	doğtaş	Ayrıntılar	183	173	20000	10000	Düzenle Sil
2	pense	ev eşyası	bosh	Ayrıntılar	43	43	55	30	Düzenle Sil
3	derin dondurucu	ev eşyası	arçelik	Ayrıntılar	269	251	18000	9000	Düzenle Sil
4	biblo	aksesuar	yeşil aksesuar	Ayrıntılar	118	118	45	30	Düzenle Sil
7	çay bardağı	mutfak aletleri	pasabahce	Ayrıntılar	825	825	250	120	Düzenle

#### Açıklama

İşlemci ürünler üzerinde düzenleme, silme veya yeni bir ürün ekleme yapabilir. Aynı şekilde marka ve kategoride ekleyebilir. Yukarıdaki siyah panel ürün ekleme panelidir aynı şekilde diğer ekleme düzenleme ve silme panelleri de benzer yapıdadır. "A-Z" Alfabe göre baştan sona sıralarken "Z-A" tersten sıralar. Ürün ara kısmında ise ürün adına göre ürün aranabilir. Sağ ve solda ise toplam kategori ve ürün sayısı belirtilmektedir. **ekle.component.html** dan alındı.



### 5.3.3 Hazırlanan Siparişler



#### Açıklama

Takvim de ayın bütün günleri sıralanmakta ve geçmiş olan günler gri , geçmiş ve sipariş olmuş günler mavi, ve geçmemiş ve siparişi varsa yeşil şeklinde gözükmetedir. Siparişi olan bir güne tıklanınca ise o güne ait siparişleri getiriyor. Aşağıdaki panelde ise siparişler ve durumları yazılıyor sipariş iptal butonuna tıklandığında ise sipariş iptal ediliyor.

**hazirlanan.component.html dan alındı.**

### 5.3.4 Sipariş geçmişi

**Yukarıda yazıldığı ve ortak kullanıldığı için bahsedilmedi.**

## 5.3 Depo Ekranı - Depo

### 5-3-1 Depo

depo							kayıt	
Ürün İd	Ürün ad	Kategori	Marka	Adet	Firma	Tutar	Sipariş durum	İşlem
3	derin dondurucu	ev eşyası	arçelik	1	Magaza 1	4842000 TL	sipariş hazırlanıyor	<div>sipariş iptal</div> <div>Tamamlandı</div>
5	scs	ürün kalktı	arçelik	3	Magaza 1	9090 TL	sipariş hazırlanıyor	<div>sipariş iptal</div> <div>Tamamlandı</div>
3	derin dondurucu	ev eşyası	arçelik	8	Magaza 1	4842000 TL	sipariş hazırlanıyor	<div>sipariş iptal</div> <div>Tamamlandı</div>
1	koltuk takımı	ev eşyası	doğtaş	5	Magaza 1	3660000 TL	sipariş hazırlanıyor	<div>sipariş iptal</div> <div>Tamamlandı</div>
16	bıçak	mutfak aletleri	tefal	5	Magaza 1	83300 TL	sipariş hazırlanıyor	<div>sipariş iptal</div> <div>Tamamlandı</div>
1	koltuk takımı	ev eşyası	doğtaş	5	Magaza 1	3660000 TL	sipariş hazırlanıyor	<div>sipariş iptal</div> <div>Tamamlandı</div>

#### Açıklama

Hazırlanan siparişler gözükür ve depo elemanı sipariş tamamlandığında tamamlandı diye onaylar ya da bir sıkıntı varsa sipariş iptal butonu ile sipariş iptal nedenini belirterek iptal eder. **depo.component.html** dan alınmıştır.

### 5-3-2 Kayıt

Toplam Ürün Çeşit Sayısı		Ürünler				Toplam Kategori Sayısı		
13		Ürün Ara				6		
Ürün İd	Ürün ad	Kategori	Marka	Ayrıntılar	Stok	Firma	Fiyat	İşlem
1	koltuk takımı	ev eşyası	doğtaş	Ayrıntılar	173	AH firması	20000	adet ekle
2	pense	ev eşyası	bosh	Ayrıntılar	43	AH firması	55	adet ekle
3	derin dondurucu	ev eşyası	arçelik	Ayrıntılar	251	AH firması	18000	adet ekle
4	biblo	aksesuar	yeşil aksesuar	Ayrıntılar	118	AH firması	45	adet ekle
7	çay bardağı	mutfak aletleri	paşabahçe	Ayrıntılar	825	AH firması	250	adet ekle
15	kağıt	kırtansiye/ofis	ve-ge	Ayrıntılar	58	AH firması	25	adet ekle
16	bıçak	mutfak aletleri	tefal	Ayrıntılar	233	AH firması	350	adet ekle
17	soyacak	mutfak aletleri	paşabahçe	Ayrıntılar	538	AH firması	50	adet ekle
18	kesme tahtası	mutfak aletleri	yeşil aksesuar	Ayrıntılar	226	AH firması	50	adet ekle

#### Açıklama

Ürün arttırımı bu panelde yaptırılır. **kayit.component.html** dan alınmıştır.

## 5.5 Mağaza Ekranı - Mağaza

### 5.5.1 Ürünler ve Mağaza alanları

Toplam Ürün Çeşit Sayısı

13

Ürünler

Ürün Ara

A-ZZ-A

Toplam Kategori Sayısı

6

Ürün Id	Ürün ad	Kategori	Marka	Ayrıntılar	Stok	Firma	Fiyat	İşlem
1	koltuk takımı	ev eşyası	doğtaş	Ayrıntılar	173	AH firması	20000	Sepete Ekle
2	pense	ev eşyası	bosh	Ayrıntılar	43	AH firması	55	Sepete Ekle
3	derin dondurucu	ev eşyası	arçelik	Ayrıntılar	251	AH firması	18000	Sepete Ekle
4	biblo	aksesuar	yeşil aksesuar	Ayrıntılar	118	AH firması	45	Sepete Ekle
7	çay bardağı	mutfak aletleri	paşabahçe	Ayrıntılar	825	AH firması	250	Sepete Ekle
15	kağıt	kırtansiye/ofis	ve-ge	Ayrıntılar	58	AH firması	25	Sepete Ekle
16	bıçak	mutfak aletleri	tefal	Ayrıntılar	233	AH firması	350	Sepete Ekle
17	soyacak	mutfak aletleri	paşabahçe	Ayrıntılar	538	AH firması	50	Sepete Ekle
18	kesme tahtası	mutfak aletleri	yeşil aksesuar	Ayrıntılar	226	AH firması	50	Sepete Ekle
19	çatal	mutfak aletleri	karaca	Ayrıntılar	237	AH firması	25	Sepete Ekle

## Açıklama

Mağaza ekranı 4 kısımdan oluşmaktadır. Ürünler, Sepet, Siparişler, Sipariş geçmişi. Ürünler mağaza bu ekranda istediği ürünü stoku aşmayacak şekilde sepet ekranına gönderebilir. Sepet ekranı siparişi onaylar ve eğer ürün stokundan fazla sipariş varsa ilk baştaki siparişleri alır fakat sonrakileri almaz. Siparişler kısmı sipariş işlemci tarafından onaylanmadıysa iptal edilebilir. Sipariş geçmişi ise mağazanın sipariş geçmişini gösterir.

**magaza.component.html alınmıştır.**

## 5.5.2 Sepet

ürünler

sepetim

siparişlerim

Sipariş geçmişi

Ürün id	Ürün ad	Kategori	Marka	stok	sipariş adeti	Firma	Fiyat	İşlem
1	koltuk takımı	ev eşyası	doğtaş	173	<input type="text" value="10"/>	AH firması	20000	<div><div>satın al</div><div>iptal et</div></div>
<div>hepsini satın al</div>								

## Açıklama

Mağaza sepete eklediği ürünleri bu ekranda görür ve adet sayısını değiştirebilir veya sepetten kaldırabilir aynı zamanda sepeti de onaylayabilir mağaza hesaptan çıksa bile ürünleri sepette kalmaya devam eder. **sepet.component.html** dan alınmıştır.

### 5.5.3 Siparişler

ürünler			sepetim		siparişlerim		Sipariş geçmişi	
Ürün id	Ürün ad	Kategori	Marka	Adet	Firma	Tutar	Sipariş durum	İşlem
3	derin dondurucu	ev eşyası	arçelik	1	AH firması	4842000 TL	sipariş hazırlanıyor	Kaldır
5	scs	ürün kalktı	arçelik	3	AH firması	9090 TL	sipariş hazırlanıyor	Kaldır
3	derin dondurucu	ev eşyası	arçelik	8	AH firması	4842000 TL	sipariş hazırlanıyor	Kaldır
1	koltuk takımı	ev eşyası	doğtaş	5	AH firması	3660000 TL	sipariş hazırlanıyor	Kaldır
1	koltuk takımı	ev eşyası	doğtaş	2	AH firması	3660000 TL	sipariş bekliyor	Kaldır
1	koltuk takımı	ev eşyası	doğtaş	2	AH firması	3660000 TL	sipariş bekliyor	Kaldır
1	koltuk takımı	ev eşyası	doğtaş	1	AH firması	3660000 TL	sipariş bekliyor	Kaldır
16	bıçak	mutfak aletleri	tefal	5	AH firması	83300 TL	sipariş hazırlanıyor	Kaldır
1	koltuk takımı	ev eşyası	doğtaş	5	AH firması	3660000 TL	sipariş hazırlanıyor	Kaldır
3	derin dondurucu	ev eşyası	arçelik	10	AH firması	4842000 TL	sipariş bekliyor	Kaldır
4	biblo	aksesuar	yeşil aksesuar	3	AH firması	5310 TL	sipariş bekliyor	Kaldır
7	derin dondurucu	ev eşyası	arçelik	3	AH firması	1610000 TL	sipariş bekliyor	Kaldır

### Açıklama

Burada mağaza siparişlerini görür fakat sipariş hazırlanma aşamasındaysa iptal edemez ama eğer hala bekletiliyorsa iptal edebilir. **siparisler.component.html** dan alınmıştır

### 5.5.4 Sipariş Geçmişi

**Yukarıda yazıldığı ve ortak kullanıldığı için bahsedilmedi.**

## 6.HTML kodları

**Aşağıda Tasarım için kullanılan HTML kodları verilmektedir sayfalar olarak ayrılmıştır.**

### -----navbar.component.html-----

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.10.3/font/bootstrap-icons.css">
```

```
<nav class="navbar w-100 d-flex py-0 align-items-center navbar-expand-md ">
```

```
<div class="container">
```

```

        <a class="navbar-brand m-0" href="#"></a>

<ul *ngIf="router.url === '/' || router.url==='/#details' ||
router.url==='/#cards' || router.url === '/#footer'" class="navbar-nav mx-auto mt-
lg-0">

    <li class="nav-item mx-auto">

        <a class="nav-link mx-3 fw-semibold text-white rounded-5 active
hov text" href="#" aria-current="page"><span class="i1">Anasayfa</span><span
class="i2"><i class="bi bi-house-door-fill"></i></span></a>

        </li>

    <li class="nav-item mx-auto">

        <a class="nav-link mx-3 fw-semibold hov rounded-5 text-white
text" href="#details"><span class="i1">Detaylar</span><span class="i2"><i
class="bi bi-question-lg"></i></span></a>

        </li>

    <li class="nav-item mx-auto ">

        <a class="nav-link mx-3 fw-semibold hov rounded-5 text-white
text" href="#footer" ><span class="i1">İletişim</span><span class="i2"><i
class="bi bi-telephone-fill"></i></span></a>

        </li>

</ul>

<form *ngIf="router.url === '/' || router.url==='/#details' ||
router.url==='/#cards' || router.url === '/#footer'" class="d-flex my-2 my-lg-0 ">

    <button (click)="ac()" (click)="handleClick($event)" id="logButton"
class="btn btn-success border-0 text-white mx-auto fw-semibold rounded-5 px-5
py-2 hov my-2 my-sm-0" type="submit"><span class="i1">Giriş Yap</span><span
class="i2"><i class="bi bi-box-arrow-in-right"></i></span></button>

</form>

```

```
    <form *ngIf="router.url !== '/#footer' && router.url !== '/' &&
router.url !== '/#details' && router.url !== '/#cards' " class="d-flex my-2 my-lg-0
">
```

```
        <button (click)="logout()" (click)="handleClick($event)"
id="logButton" class="btn btn-success border-0 text-white mx-auto fw-semibold
rounded-5 px-5 py-2 hov my-2 my-sm-0" type="submit"><span class="i1">Çıkış
Yap</span><span class="i2"><i class="bi bi-box-arrow-in-
right"></i></span></button>
```

```
    </form>
```

```
</div>
```

```
</nav>
```

```
<div id="login" class="d-none card p-3 position-absolute border-0 mx-auto
rounded-5 ">
```

```
    <i id="cross" (click)="kapat()" (click)="handleClick($event)" class="bi bi-x-
lg cross"></i>
```

```
    <h5 id="actLOG" (click)="gir()" (click)="handleClick($event)" class="act d-
inline-block text-white px-4 py-1 rounded-5 c fs-5">Giriş</h5>
```

```
    <div id="log" class=" py-5 p-25 rounded-5">
```

```
        <div id="logi">
```

```
            <form class=" p-3 rounded-3" [formGroup]="loginForm"
(ngSubmit)="loginSubmitted(">
```

```
                <div class="row">
```

```
                    <div class="col mb-3">
```

```
                        <label>
```

```
                            E-mail adresi
```

```
                        </label>
```

```
                            <input type="email" class="border-0 rounded-5 in"
id="email" formControlName="email">
```

```
                            <div class="form-text text-danger"
```

```
*ngIf="email.errors?.['required']&&(email.dirty||email.touched)">
```

```
                                email boş bırakılamaz
```

```

        </div>

        <div class="form-text text-danger"

*ngIf="email.errors?.['email']&&(email.dirty||email.touched)">

            email ****@***.*** gibi olmalı

        </div>

    </div>

    <div class="row">

        <div class="col mb-3">

            <label for="pass" class="form-label"> Password</label>

            <input type="password" class="border-0 rounded-5 in "
id="pass" formControlName="pass">

            <div class="form-text text-danger"

*ngIf="pass.errors?.['required']&&(pass.dirty||pass.touched)">

                şifre boş bırakılamaz

            </div>

            <div class="form-text text-danger"

*ngIf="pass.errors?.['minlength']&&(pass.dirty||pass.touched)">

                şifre fazla kısa<br> uzunluğu 6 ile 15 arası olmalı.

            </div>

        </div>

    </div>

    <div class="row ">

        <div class="col-6 offset-3 ">

            <button type="submit" [disabled]="loginForm.invalid"
class="btn btn-secondary me-auto px-4 text-white" id="email">Login</button>

        </div>

    </div>

</form>

```

```
</div>
</div>
</div>
<div id="blocker" class="blocker d-none"></div>
```

---

#### ----content.component.html----

```
<navbar></navbar>
<div class="cont-bg">
  <div id="carouselId" class="carousel slide shadow overflow-hidden w-100 mb-2
" data-bs-ride="carousel">
    <ol class="carousel-indicators">
      <li data-bs-target="#carouselId" data-bs-slide-to="0" class="active"
aria-current="true" aria-label="First slide"></li>
      <li data-bs-target="#carouselId" data-bs-slide-to="1" aria-
label="Second slide"></li>
      <li data-bs-target="#carouselId" data-bs-slide-to="2" aria-
label="Third slide"></li>
    </ol>
    <div class="carousel-inner position-relative " role="listbox">
      <div class="carousel-item active">
        
        <div class="cor-text">
          <h5 class="my-2">DMA'ya Hoşgeldiniz!</h5>
          <p>DMA hizmeti birçok farklı cihazdan ve ağdan sanal
veritabanları sayesinde bilgi yönetimini kolay ve hızlı bir şekilde uygulamasını
sağlıyor!</p>
          <a href="#details"> <button class="cor-but btn btn-primary
text-white px-2" >Ayrıntılar için tıkla!</button> </a>
        </div>
      </div>
      <div class="carousel-item ">
```



```


<div class="cor-text">

    <h5 class="my-2">DMA hizmetinin avantajları çok!</h5>

    <p>DMA'nın birçok avantajı var işletmeni sen de bir adım öne
taşımak istiyorsan tıkla!</p>

    <a href="#cards"> <button class="cor-but btn btn-secondary text-
white px-2">Avantajlar için tıkla!</button> </a>

</div>

</div>

<div class="carousel-item">

    <div class="cor-text ">

        <h5 class="my-2">Sizde Katılın!</h5>

        <p>DMA hizmetinden yararlanmak bir tık uzağında!</p>

        <a href="#footer"> <button class="cor-but btn btn-danger text-
white px-2">Bizim ile iletişime geç!</button> </a>

    </div>

</div>

    <button class="carousel-control-prev" type="button" data-bs-
target="#carouselId" data-bs-slide="prev">

        <div class="left"><span class="carousel-control-prev-icon" aria-
hidden="true"></span></div>

        <span class="visually-hidden">Previous</span>

    </button>

    <button class="carousel-control-next" type="button" data-bs-
target="#carouselId" data-bs-slide="next">

        <div class="right"><span class="carousel-control-next-icon" aria-
hidden="true"></span></div>

        <span class="visually-hidden">Next</span>

    </button>

</div>
```

</div>

```
<div id="details" class="w-75 px-5 mt-5 overflow-hidden rounded-5 py-sm-3 py-
lg-5 mx-auto">
```

```
<div class="d-flex flex-wrap mx-0 px-0">
```

```
<div class="col-lg-7 col-sm-12 px-sm-5">
```

```

```

```
</div>
```

```
<div class="col-lg-5 col-sm-12">
```

```
<div class="card-body justify-content-center align-items-center py-
lg-5 text-center">
```

```
<h4 class="card-title h1 text-info py-4"><span class="text-
success">DMA</span> Nedir?</h4>
```

```
<p class="lead ">DMA bir ticari amaç güden işletmeler için verilen bir
hizmettir.DMA hizmeti,iş ile alakalı olan
```

```
verilerinizi sanal bir veritabanında tutar ve güncel olarak
bilgileri işler bu sanal veritabanında isteğiniz şekilde yönetmenizi ve akıllı
sistemi ile işletmenizi bir adım ileri taşımayı amaçlar. </p>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<h2 class="text-center text-success py-5">DMA'nın Avantajları!</h2>
```

```
<div id="cards" class=" mx-0 py-2 text-white cardsAV">
```

```
<div class=" rounded-5 bg-primary text-center">
```

```
<i class="bi bi-database-check fs-1"></i><br>
```

```
<h4 class="py-3">Sanal Veritabanı</h4>
```

```
<p>Sanal veritabanları ile internetiniz olduğu her anda ulaşabilirlik
sağlıyor ve size daha kısa sürelerde isteklerinizi gerçekleştirebiliyoruz.</p>
```

```
</div>

<div class="rounded-5 bg-secondary text-center">

    <i class="bi bi-universal-access fs-1"></i><br>

    <h4 class="py-3">Erişebilirlik</h4>

    <p>DMA'da erişebilirlik tam seviyede! İstedığınız şekilde veritabanını
düzenleyebilir veya altınızda çalışan işçileri yetkilendirebilirsiniz!</p>

</div>

<div class="rounded-5 bg-primary text-center">

    <i class="bi bi-emoji-smile fs-1"></i><br>

    <h4 class="py-3">Akıllı Sistem</h4>

    <p>DMA hizmeti istitaki veriler üzerinden size çeşitli tavsiyeler
verir ve bir nevi sizin küçük bir yardımcınız olur.</p>

</div>

<div class="rounded-5 bg-secondary text-center">

    <i class="bi bi-bar-chart-fill fs-1"></i><br>

    <h4 class="py-3">İstatiki Veri Analizi</h4>

    <p>DMA hizmeti Veritabanındali verileri çeşitli işlemlerden geçirerek
size kolay anlaşılabilir grafik ve tablolarla size sunar.</p>

</div>

</div>

<div id="footer" class="d-flex flex-wrap text-center g-0 px-0 py-4 mx-0 text-
white mt-5 ">

    <div class="col-lg-3 frm mt-0 mx-auto px-3 col-md-4 col-sm-8 col-12 order-
1">

        <h3>Logo</h3>

    </div>
```

```
<div class="col-lg-3 mx-auto frm mt-5 mt-md-5 mt-lg-0 px-3 col-12 order-3">
```

```
<h3>Hakkımızda!</h3>
```

```
<p>DMA projesini 2 kişi geliştirmiştir bu iki kişi şuanda Dokuz Eylül Bil. Prog. bölümünde aktif olarak okumaktadır.Tasarım kısmı Umut Güngör,Arka İşlem kısmını Aytaç Melih Kılıç üstlenmiştir.DMA projesi Okul projesi olarak sunulmak için hazırlanmıştır. </p>
```

```
</div>
```

```
<div class="col-lg-6 mx-auto my-lg-0 px-3 col-md-8 col-12 order-2">
```

```
<div class="frm w-100 mx-0 text-center my-0">
```

```
<h3 class="pt-3">İletişim Bilgileri</h3>
```

```
<h5 class="py-2">E-posta</h5>
```

```
<span class="bg-primary p-2 px-4 rounded-5">  
<u>dma.info.tr@gmail.com</span><br><br>
```

```
<h5 class="py-2" >Tel</h5>
```

```
<span class="bg-primary text-decoration-none p-2 px-4 rounded-5">+90 534 035 2964</span><br><br>
```

```
<span class="bg-primary text-decoration-none p-2 px-4 rounded-5">+90 545 287 57 91</span>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
-----  
----yonetici.component.html----
```

```
<navbar></navbar>
```

```
<div class="w-100">
```

```
<ul id="list" class="main-opt list-unstyled my-0 w-100 text-white">
```

```
<li id="0" routerLink="/main" routerLinkActive="Sitem" [routerLinkActiveOptions]="{exact : true}"  
class="list-item " >Grafik ve Analiz</li>
```

```
<li id="1" routerLink="/main/raport" class="list-item" [routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >Raporlar</li>
```

```
<li id="2" routerLink="/main/recently" class="list-item"[routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >Sipariş Geçmişi</li>
```

```
<li id="3" routerLink="/main/hesap" class="list-item"[routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >hesap</li>
```

```
</ul>
```

```
</div>
```

```
<div id="carouselId" class="carousel car text-center text-white card-bg slide shadow py-2 mt-lg-3 mt-1 " data-
bs-ride="carousel">
```

```
<ol class="carousel-indicators">
```

```
<li data-bs-target="#carouselId" data-bs-slide-to="0" class="active" aria-current="true" aria-label="First
slide"></li>
```

```
<li data-bs-target="#carouselId" data-bs-slide-to="1" aria-label="Second slide"></li>
```

```
<li data-bs-target="#carouselId" data-bs-slide-to="2" aria-label="Third slide"></li>
```

```
<li data-bs-target="#carouselId" data-bs-slide-to="3" aria-label="Fourth slide"></li>
```

```
<li data-bs-target="#carouselId" data-bs-slide-to="4" aria-label="Fiveth slide"></li>
```

```
<li data-bs-target="#carouselId" data-bs-slide-to="5" aria-label="Sixth slide"></li>
```

```
<li data-bs-target="#carouselId" data-bs-slide-to="6" aria-label="Seventh slide"></li>
```

```
<li data-bs-target="#carouselId" data-bs-slide-to="7" aria-label="Eight slide"></li>
```

```
<li data-bs-target="#carouselId" data-bs-slide-to="8" aria-label="Nineth slide"></li>
```

```
</ol>
```

```
<div class="carousel-inner h-100 w-100 position-relative " role="listbox">
```

```
<div class="carousel-item px-4 h-100 w-100 active">
```

```
<h5 class="mt-4 mb-3"><span class="text-secondary">{{tavsiyelerCopy[1]}}</span> Satış Artışı!</h5>
```

<p class="mt-4"><span class="text-secondary">{{tavsiyelerCopy[1]}}</span> adlı ürün <span class="guzel1">en çok</span> satış yapan ürün daha fazla yere ihraç edip daha fazla kâr elde edebilirsiniz!</p>  
</div>

<div class="carousel-item px-4 h-100 w-100">  
<h5 class="mt-4 mb-3"><span class="text-secondary">{{tavsiyelerCopy[3]}}</span> kategorisi en çok ilgi gören kategoriniz!</h5>  
<p class="mt-4"><span class="text-secondary">{{tavsiyelerCopy[3]}}</span> kategorisi en çok satın alınan kategoriniz bu kategoriye daha fazla ürün ekleyebilirsiniz!</p>  
</div>

<div class="carousel-item px-4 h-100 w-100">  
<h5 class="mt-4 mb-3"><span class="text-success">{{tavsiyelerCopy[5]}}</span> markası en çok ilgi gören marka!</h5>  
<p class="mt-4"><span class="text-success">{{tavsiyelerCopy[5]}}</span> markası en çok satın alınan kategoriniz bu markadan daha fazla ürün talep edebilirsiniz!</p>  
</div>

<div class="carousel-item px-4 h-100 w-100">  
<h5 class="mt-4 mb-3"><span class="guzel2">{{tavsiyelerCopy[7]}}</span> markası en çok ilgi gören marka!</h5>  
<p class="mt-4"><span class="guzel2">{{tavsiyelerCopy[7]}}</span> markası en çok satın alınan kategoriniz bu markadan daha fazla ürün talep edebilirsiniz!</p>  
</div>

<div class="carousel-item px-4 h-100 w-100">  
<h5 class="mt-4 mb-3">İki haftadır satmayan ürün veya ürünleriniz <span class="guzel1">{{ikihafta[0] == "" ? "yok" : ikihafta[0]}}!</span></h5>  
<p class="mt-4"><span class="guzel1">{{ikihafta[0] == "" ? "satılmayan ürün yok mükemmmel" : "satılmayan ürün veya ürünler "+ikihafta[0]}}!</span></p>  
</div>

```
<div class="carousel-item px-4 h-100 w-100">

  <h5 class="mt-4 mb-3"><span class="text-danger">En az satan</span> ürününüz!</h5>

  <p class="mt-4">En az satan ürününüz <span class="text-danger">{{ tavsiyelerCopy[2]
}}</span>!</p>

</div>
```

```
<div class="carousel-item px-4 h-100 w-100">

  <h5 class="mt-4 mb-3"><span class="text-success">En az satan</span> kategoriniz!</h5>

  <p class="mt-4">En az satan kategoriniz <span class="text-
success">{{ tavsiyelerCopy[4] }}</span>!</p>

</div>
```

```
<div class="carousel-item px-4 h-100 w-100">

  <h5 class="mt-4 mb-3"><span class="guzel1">En az satan</span> markanız!</h5>

  <p class="mt-4">En az satan markanız <span class="guzel1">{{ tavsiyelerCopy[6]
}}</span>!</p>

</div>
```

```
<div class="carousel-item px-4 h-100 w-100">

  <h5 class="mt-4 mb-3"><span class="text-danger">En az satış oranına</span> sahip
ürünüz!</h5>

  <p class="mt-4">En az satış oranına sahip ürününüz <span class="text-
danger">{{ tavsiyelerCopy[8] }}</span>!</p>

</div>
```

```
</div>
```

```
<button class="carousel-control-prev" type="button" data-bs-target="#carouselId" data-bs-slide="prev">

  <div class="left"><span class="carousel-control-prev-icon" aria-hidden="true"></span></div>

  <span class="visually-hidden">Previous</span>

</button>
```

```
<button class="carousel-control-next" type="button" data-bs-target="#carouselId" data-bs-slide="next">

  <div class="right"><span class="carousel-control-next-icon" aria-hidden="true"></span></div>

  <span class="visually-hidden">Next</span>

</button>

</div>
```

```
<div class="bg mt-lg-3 mt-1 ">

<div id="blocks" class=" text-white font">

<div id="graph" class="h-100 card-bg px-md-4 pb-sm-3 px-3 ">

  <div class="py-2 pe-4 align-items-center mt-3 d-flex gap-2" id="group">

    <div class="col-4 gap-2">

      <div class="d-flex gap-2">

        <button id="st" (click)="tip='bar'" (click)="this.btnClick($event)" class=" btn col-6 h-50 btn-
primary">Sütun</button>

        <button id="ln" (click)="tip='line'" (click)="this.btnClick($event)" class="btn col-6 h-50 text-white btn-
secondary">Çizgi</button>

      </div>

    </div>
```

```
<div class="d-flex gap-1">

  <button id="LastYear" (click)="this.lastYear($event)" class="btn mt-2 col-3 text-white btn-sm btn-
success">Son Yıl</button>

  <button id="LastMonth" (click)="this.lastMonth($event)" class="btn mt-2 col-3 text-white btn-sm btn-
info">Son Ay</button>

  <button id="LastWeek" (click)="this.lastWeek($event)" class="btn mt-2 col-3 text-white btn-sm btn-
warning">Son Hafta</button>

  <button id="today" (click)="this.todayy($event)" class="btn mt-2 col-3 text-white aktif btn-sm btn-
primary">Bugün</button>

</div>
```

```
</div>
```

```
<h4 id="baslik" class="col-4 m-0 p-0 text-center d-inline-block">En Çok Satan ilk 10 Ürün</h4>

<div class="col-4 text-center">
```



```
<select (change)="val($event)" class="rounded-5 px-2 in" id="selectMod">
  <option value="satan">En Çok Satan ilk 10 Ürün</option>
  <option value="sepet">En Çok Sepette Bulunan Ürünler ilk 10</option>
  <option value="kar">En Çok Kâr Edilen ilk 10 Ürün</option>
</select>

<button id="ozel" (click)="acc($event)" class="btn btn-success border-0 mt-3 w-50 text-white border-none">Özel Tarih</button>

</div>

</div>

<canvas class="py-2" id="chart"></canvas>

<div class="d-flex text-center">

  <div class="col-6">
    Kategori<br>
    <select (change)="filtre()" class="rounded-5 w-75 overflow-y-auto px-2 in" id="selectCat">
      <option value="0">Hepsi</option>
      <ng-container *ngFor="let kat of kategoriler">
        <option *ngIf="kat.kategori_ID>=1" [value]="kat.kategori_ID">{{ kat.kategori_AD }}</option>
      </ng-container>

      <!--ngfor-->

    </select>
  </div>

  <div class="col-6">
    Marka<br>
    <select (change)="filtre()" class="rounded-5 overflow-y-auto w-75 px-2 in" id="selectMark">
      <option value="0">Hepsi</option>
      <option *ngFor="let mak of markalar" value={{ mak.marka_ID }}>{{ mak.marka_AD }}</option>
      <!--ngfor-->
    </select>
  </div>
</div>
```

</select>

</div>

<div>

</div>

</div>

</div>

<div id="analysis" class="scroll scroll-1 overflow-y-auto card-bg ">

<div class="text-center">

<h4 class="p-2 ">Grafik Analizleri</h4>

<button (click)="cokAz()" id="cokaz" class="aktif btn btn-sm me-2 f-but btn-primary">Çoktan aza</button>

<button (click)="azCok()" id="azcok" class="btn btn-sm f-but text-white btn-secondary">Azdan çok</button>

<h5 class="pt-4 pb-2">Kısaltma Açıklamaları</h5>

<div \*ngIf="this.deger=='satan'">

<div \*ngIf="this.zaman=='today'">

<span class="mt-1">BGSA : Bu günki satış adeti </span><!--Satış Oranı = (Bu Haftaki Satış Tutarı / Geçen Haftaki Satış Tutarı) \* 100-->

</div>

<div \*ngIf="this.zaman=='week'">

<span class="mt-1">BHSA : Bu haftaki satış adeti </span><!--Satış Oranı = (Bu Haftaki Satış Tutarı / Geçen Haftaki Satış Tutarı) \* 100-->

</div>

<div \*ngIf="this.zaman=='month'">

<span class="mt-1">BASA : Bu ayki satış adeti </span><!--Satış Oranı = (Bu Haftaki Satış Tutarı / Geçen Haftaki Satış Tutarı) \* 100-->

</div>

```
<div *ngIf="this.zaman=='year'">

  <span class="mt-1" >BYSA : Bu yılki satış adeti </span><!--Satış Oranı = (Bu Haftaki Satış Tutarı / Geçen
Haftaki Satış Tutarı) * 100-->
```

```
</div>
```

```
</div>
```

```
<div *ngIf="this.deger=='sepet'">
```

```
  <div *ngIf="this.zaman=='today'">
```

```
    <span class="my-1">BGSÜ : bugünkü sepetteki ürünler </span><br>
```

```
  </div>
```

```
  <div *ngIf="this.zaman=='week'">
```

```
    <span class="my-1">BHSÜ : Bu haftaki sepetteki ürünler</span><br>
```

```
  </div>
```

```
  <div *ngIf="this.zaman=='month'">
```

```
    <span class="my-1">BASÜ : Bu ayki sepetteki ürünler</span><br>
```

```
  </div>
```

```
  <div *ngIf="this.zaman=='year'">
```

```
    <span class="my-1">BYSÜ : Bu yılki sepetteki ürünler</span><br>
```

```
  </div>
```

```
</div>
```

```
<div *ngIf="this.deger=='kar'">
```

```
  <div *ngIf="this.zaman=='today'">
```

```
    <span class="my-1">BGK : Bugünkü kar </span><br>
```

```
  </div>
```

```
  <div *ngIf="this.zaman=='week'">
```

```
    <span class="my-1">BHK : Bu haftaki kar </span><br>
```

```
  </div>
```

```
  <div *ngIf="this.zaman=='month'">
```

```
    <span class="my-1">BAK : Bu ayki kar </span><br>
```

</div>

<div \*ngIf="this.zaman=='year'">

<span class="my-1">BYK : Bu yılki kar </span><br>

</div>

</div>

</div><br>

<div class="text-center">

Ürün Ara<br>

<input type="text" (change)="ara(\$event)" class="w-50 in rounded-5 border border-0">

</div>

<table class="table table-borderless text-center my-4 text-white">

<thead \*ngIf="this.deger=='satan'" class="text-center">

<tr \*ngIf="this.zaman=='today'">

<th class="text" scope="col">Ürün id</th>

<th class="text" scope="col">Ürün ad</th>

<th class="text" scope="col">BGSA</th>

</tr>

<tr \*ngIf="this.zaman=='week'">

<th class="text" scope="col">Ürün id</th>

<th class="text" scope="col">Ürün ad</th>

<th class="text" scope="col">BHSO</th>

</tr>

<tr \*ngIf="this.zaman=='month'">

<th class="text" scope="col">Ürün id</th>

<th class="text" scope="col">Ürün ad</th>

<th class="text" scope="col">BASA</th>

</tr>

<tr \*ngIf="this.zaman=='year'">

<th class="text" scope="col">Ürün id</th>

<th class="text" scope="col">Ürün ad</th>

```
<th class="text" scope="col">BYSA</th>

</tr>

</thead>

<thead *ngIf="this.deger=='sepet'" class="text-center">

  <tr *ngIf="this.zaman=='today'">

    <th class="text" scope="col">Ürün id</th>

    <th class="text" scope="col">Ürün ad</th>

    <th class="text" scope="col">BGSÜ</th>

  </tr>

  <tr *ngIf="this.zaman=='week'">

    <th class="text" scope="col">Ürün id</th>

    <th class="text" scope="col">Ürün ad</th>

    <th class="text" scope="col">BHSÜ</th>

  </tr>

  <tr *ngIf="this.zaman=='month'">

    <th class="text" scope="col">Ürün id</th>

    <th class="text" scope="col">Ürün ad</th>

    <th class="text" scope="col">BASÜ</th>

  </tr>

  <tr *ngIf="this.zaman=='year'">

    <th class="text" scope="col">Ürün id</th>

    <th class="text" scope="col">Ürün ad</th>

    <th class="text" scope="col">BYSÜ</th>

  </tr>

</thead>

<thead *ngIf="this.deger=='kar'" class="text-center">

  <tr *ngIf="this.zaman=='today'">

    <th class="text" scope="col">Ürün id</th>

    <th class="text" scope="col">Ürün ad</th>
```

```

        <th class="text" scope="col">Kar Ort.</th>

        <th class="text" scope="col">BGK</th>

    </tr>

    <tr *ngIf="this.zaman=='week'">

        <th class="text" scope="col">Ürün id</th>

        <th class="text" scope="col">Ürün ad</th>

        <th class="text" scope="col">Kar Ort.</th>

        <th class="text" scope="col">BHK</th>

    </tr>

    <tr *ngIf="this.zaman=='month'">

        <th class="text" scope="col">Ürün id</th>

        <th class="text" scope="col">Ürün ad</th>

        <th class="text" scope="col">Kar Ort.</th>

        <th class="text" scope="col">BAK</th>

    </tr>

    <tr *ngIf="this.zaman=='year'">

        <th class="text" scope="col">Ürün id</th>

        <th class="text" scope="col">Ürün ad</th>

        <th class="text" scope="col">Kar Ort.</th>

        <th class="text" scope="col">BYK</th>

    </tr>

</thead>

```

<!-- ilk hafta dolmadan hiç bir veri gösterilemez ve ilk haftada total stoğun ne kadarı satıldıysa ona göre oran alınacak

sonraki haftalarda karşılaştırma yapılacak.-->

```

<tbody *ngIf="deger=='satan' || deger=='sepet'" class="w-100">

    <tr *ngFor="let urun of analiztab1"><!--ngFor-->

        <td>#{{ urun.id }}</td>

        <td>{{ urun.ad }}</td>

        <td>{{ urun.adet }}</td>

```

```
</tr>

</tbody>

<tbody *ngIf="deger=='kar'" class="w-100">

  <tr *ngFor="let urun of analiztab1"><!--ngFor-->

    <td>#{ { urun.id } }</td>

    <td>{ { urun.ad } }</td>

    <td>{ { (((urun.satis-urun.mal)/urun.satis)*100).toFixed(2) } }%</td>

    <td>{ { urun.kar } }TL</td>

  </tr>

</tbody>

</table>

</div>

<div id="login" class="d-none login p-3 border-0 mx-auto rounded-5">

  <i id="cross" (click)="kapatt($event)" class="bi bi-x-lg cross"></i>

  <div class="log py-5 p-25 rounded-5">

    <h5 id="date" class="act d-none text-center text-white px-4 py-1 rounded-5 mb-3 c fs-5">Tarih</h5>

    <div id="logs">

      <h6 class="me-auto">Başlangıç Tarihi</h6>

      <input id="date-beg" class="border-0 rounded-5 in" type="date" [formControl]="startDateControl">

    </div>

    <h6 class="me-auto pt-3">Bitiş Tarihi</h6>

    <input id="date-fin" class="border-0 rounded-5 in" type="date" [formControl]="endDateControl">

  </div>

  <button class="btn mt-4 my-3 d-block rounded-5 border border-2 border-secondary fs-6 w-100 btn-outline-secondary" (click)="submitForm()">

    Tamamla!

  </button>

</div>

</div>

</div>

</div>
```

## ----yonet.rapor.component.html----

```
<navbar></navbar>

<div class="w-100">

  <ul id="list" class="main-opt list-unstyled my-0 w-100 text-white">

    <li id="0" routerLink="/main" routerLinkActive="Sitem" [routerLinkActiveOptions]="{exact : true}"
class="list-item " >Grafik ve Analiz</li>

    <li id="1" routerLink="/main/raport" class="list-item" [routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >Raporlar</li>

    <li id="2" routerLink="/main/recently" class="list-item"[routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >Sipariş Geçmişi</li>

    <li id="3" routerLink="/main/hesap" class="list-item"[routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >hesap</li>

  </ul>

</div>

<div id="content" class="font">

  <div class="mb-4 px-sm-2 my-1 pb-2 px-1 mx-lg-4 mx-md-2 mx-0 text-white ">

    <div class="d-flex gap-1 px-md-4 mb-2 px-1">

      <button id="LastYear" (click)="this.lastYear($event)" class="btn mt-3 col-3 text-white rounded-5 btn-
sm btn-success">Son Yıl</button>

      <button id="LastMonth" (click)="this.lastMonth($event)" class="btn mt-3 col-3 text-white rounded-5
btn-sm btn-info">Son Ay</button>

      <button id="LastWeek" (click)="this.a($event)" class="btn mt-3 col-3 text-white rounded-5 btn-sm btn-
warning">Son Hafta</button>

      <button id="LastDay" (click)="this.lastDay($event)" class="btn mt-3 col-3 text-white rounded-5 btn-sm
btn-primary">Bu Gün</button>

    </div>

    <div id="pdfContent" class="px-md-3 px-1 d-flex ">

      <div class="row mx-0 w-100">

        <div class=" mb-2 px-2 hei-xl col-md-6 col-12">
```



<div class="border border-0 position-relative card-bg mb-2 hei py-2 d-flex overflow-auto flex-column align-items-center justify-content-center rounded-5 ">

<h3 class="text-center position-static py-4 top-0">Stok Bilgileri</h3>

<ul class="list-unstyled ps-3 w-100 text-center"><!-- geçmiş view , alınanlar -->

<li class="list-group-item mb-2">Toplam {{totalsat}} stok azaldı.</li>

<li class="list-group-item mb-2">Toplam {{totalal}} stok yenilendi.</li>

<li class="list-group-item mb-2">En çok {{enkat}} kategorisinin stoğu azaldı.</li>

<li class="list-group-item mb-2">En çok {{enmark}} markasının stoğu azaldı.</li>

<li class="list-group-item mb-2">En çok {{satan}} ürününün stoğu azaldı.</li>

</ul>

</div>

<div class="border border-0 position-relative card-bg py-2 hei text-center hei d-flex flex-column align-items-center justify-content-center rounded-5 ">

<h3 class="text-center position-static top-0 py-4">Performans Bilgileri</h3>

<ul class="list-unstyled ps-3 w-100 text-center"><!-- geçmiş view,siparişler-->

<li class="list-group-item mb-2">Başarılı biten sipariş oranı {{ ((basarili / (basarili + basarisiz)) \* 100).toFixed(1) }}%</li>

<li class="list-group-item mb-2">İptal edilen sipariş oranı {{ ((basarisiz / (basarili + basarisiz)) \* 100).toFixed(1) }}%</li>

<li class="list-group-item mb-2">Ortalama sipariş tamamlama süresi {{teslimler}} gün.</li>

<li class="list-group-item mb-2">İşlemi biten sipariş sayısı {{totalsip}}.</li>

</ul>

</div>

</div>

<div class=" mb-2 px-2 hei-xl col-md-6 col-12">

<div class="border border-0 position-relative mb-2 card-bg py-2 d-flex hei-md overflow-auto flex-column align-items-center justify-content-center rounded-5 ">

<h3 class="text-center position-static top-0 py-4">Satış Bilgileri</h3>

```
<ul class="list-unstyled ps-3 w-100 text-center"><!--geçmiş view en az satan ?-->

<li class="list-group-item mb-2">Başarılı satış sayısı {{basarili}}.</li>

<li class="list-group-item mb-2">İptal edilen satış sayısı {{basarisiz}}.</li>

<li class="list-group-item mb-2">En çok satan ürün {{satan}}.{{satankez}} kez satıldı.</li>

<li class="list-group-item mb-2">En az satan ürün {{enazsatan}}. {{enazsatankez}} kez
satıldı.</li>

<li class="list-group-item mb-2">En çok satan kategori {{enkat}}. {{katkez}} kez satıldı.</li>

<li class="list-group-item mb-2">En az satan kategori {{enazkat}}.{{enazkatkez}} kez satıldı.(hiç
satmayanlar hariç)</li>

<li class="list-group-item mb-2">En çok satan marka {{enmark}}. {{markkez}} kez satıldı.</li>

<li class="list-group-item mb-2">En az satan marka {{enazmark}} . {{enazmarkkez}} kez
satıldı.</li>
```

```
</ul>
```

```
</div>
```

```
<div class="border border-0 position-relative hei-small card-bg py-2 d-flex overflow-auto flex-column
align-items-center justify-content-center rounded-5 ">
```

```
<h3 class="text-center position-static top-0 py-4">Finans Bilgileri</h3>
```

```
<ul class="list-unstyled ps-3 w-100 text-center"><!--geçmiş view-->
```

```
<li class="list-group-item mb-2">Toplam satış geliri {{topkaz}} TL.</li>
```

```
<li class="list-group-item mb-2">Toplam ürün maliyet {{topmal}} TL.</li>
```

```
<li class="list-group-item mb-2">Toplam kar oranı {{{(topkaz-
topmal)/(topkaz)*100).toFixed(2)}}}%.</li>
```

```
</ul>
```

```
</div>
```

```
</div>
```

```
<button (click)="PDFDownload()" class="btn btn-secondary border-2 rounded-5 d-none text-white px-
4">PDF indir</button>
```

```
</div>
```

```
</div>
```

```
-----yonet-gecmis.component.html-----
```

```

<navbar></navbar>

<div class="w-100">

  <ul id="list" class="main-opt list-unstyled my-0 w-100 text-white">

    <li id="0" routerLink="/main" routerLinkActive="Sitem" [routerLinkActiveOptions]="{exact : true}"
class="list-item " >Grafik ve Analiz</li>

    <li id="1" routerLink="/main/raport" class="list-item" [routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >Raporlar</li>

    <li id="2" routerLink="/main/recently" class="list-item"[routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >Sipariş Geçmişi</li>

    <li id="3" routerLink="/main/hesap" class="list-item"[routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >hesap</li>

  </ul>

</div>

<app-gecmis></app-gecmis>

```

#### ----yonet-hesap.component.html----

```

<navbar></navbar>

<div class="w-100">

  <ul id="list" class="main-opt list-unstyled my-0 w-100 text-white">

    <li id="0" routerLink="/main" routerLinkActive="Sitem" [routerLinkActiveOptions]="{exact : true}"
class="list-item " >Grafik ve Analiz</li>

    <li id="1" routerLink="/main/raport" class="list-item" [routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >Raporlar</li>

    <li id="2" routerLink="/main/recently" class="list-item"[routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >Sipariş Geçmişi</li>

    <li id="3" routerLink="/main/hesap" class="list-item"[routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >hesap</li>

  </ul>

</div>

<div id="pdfContent" class="px-md-3 px-1 d-flex ">

  <div class="row mx-0 mt-2 w-100">

    <div class=" mb-2 px-2 hei-md col-lg-8 col-12">

      <div class="border border-0 text-white position-relative card-bg mb-2 hei-md py-2  rounded-5 ">

```

<h3 class="my-3 d-inline-block px-5">Hesaplar</h3>

<button class="btn yen my-3 btn-sm text-white mx-4 position-absolute end-0" (click)="pan\_ac()">Personel Ekle</button>

<div class="table-responsive max-overflow1">

<table class="table table-borderless text-center text-white">

<thead>

<tr>

<th class="text" scope="col">Personel E-mail</th>

<th class="text" scope="col">Personel Şifre</th>

<th class="text" scope="col">Personel Rol</th>

<th class="text" scope="col">Personel İsim</th>

<th class="text" scope="col">İşlem</th>

</tr>

</thead>

<tbody class="w-100">

<tr class="" \*ngFor="let personel of personels">

<td class="text text-break" scope="row">{{ personel.email }}</td>

<td class="text text-break">{{ personel.pass }}</td>

<ng-container \*ngFor="let rol of roller">

<td class="text text-break" \*ngIf="rol.ls === personel.ls">{{ rol.lisans\_adlari }}</td>

</ng-container>

<td class="text text-break">{{ personel.firstLast }}</td>

<td class="text max-width text-break">

<button type="button" class="btn btn-sm text-white btn-width px-0 mx-0 btn-secondary" (click)="duzenle(personel)">Düzenle</button><br>

<button type="button" \*ngIf="personel.ls!=1" class="btn btn-width btn-sm px-0 mx-0 text-white btn-danger" (click)="sil(personel)">Sil</button>

```
        <button type="button" class="btn btn-sm text-white btn-width px-0 mx-0 btn-primary"
(click)="hesap(personel)">Hesap Hareketleri</button>
```

```
    </td>
```

```
</tr>
```

```
</tbody>
```

```
</table>
```

```
</div>
```

```
</div>
```

```
</div>
```

```
<div class=" mb-2 px-2 hei-md col-lg-4 col-12">
```

```
<div class="border border-0 text-white position-relative card-bg py-2 hei-md text-center d-flex flex-column
align-items-center justify-content-center rounded-5 ">
```

```
    <button class="btn btn-sm text-white rounded-5 btn-success" (click)="hepsi()">Tüm Hesap
Hareketleri</button>
```

```
<div class="h-50 d-flex flex-column align-items-center justify-content-center ">
```

```
    <h3 class="text-center py-2">Toplam Hesap Sayısı</h3>
```

```
    <h5 class="px-5 py-2 text-white d-inline-block rounded-5 bg-secondary">{{personelsay}}</h5>
```

```
</div>
```

```
<div class="h-50 d-flex flex-column align-items-center justify-content-center">
```

```
<div class="table-responsive max-overflow1">
```

```
<table class="table table-borderless text-center text-white">
```

```
<thead>
```

```
<tr>
```

```
    <th class="text" scope="col">Personel İsim</th>
```

```
    <th class="text" scope="col">Personel Hesap aktifliği (günlük)</th>
```

```
    <th class="text" scope="col">Personel Hesap aktifliği (haftalık)</th>
```

```

</tr>

</thead>

<tbody class="w-100">

<tr class="" *ngFor="let giris of girislerCopy">

  <td class="text text-break" scope="row">{{ giris.firstLast }}</td>

  <td class="text text-break">{{ giris.dakika_1gun }} dk</td>

  <td class="text text-break">{{ giris.dakika_7gun }} dk</td>

</tr>


</tbody>

</table>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

</div>

<form [formGroup]="personelForm" >

  <div id="pro-add" class="d-none login p-3 my-auto border-0 mx-auto rounded-5">

    <i id="cross" (click)="pan_kapat()" class="bi bi-x-lg cross"></i>

    <div id="pro-form" class="log py-5 p-25 rounded-5">

      <div id="pro-frm">

        <h6 class="me-auto">Personel E-mail</h6>

        <input id="pro-id" formControlName="personel_email" required maxlength="50" minlength="3"
class="border-0 mb-2 rounded-5 in" type="text">

        <div *ngIf="personelForm.get('personel_email')!.invalid &&
(personelForm.get('personel_email')!.dirty || personelForm.get('personel_email')!.touched)">

```

```
<div *ngIf="personelForm.get('personel_email')!.hasError('required')">E-mail alanı boş  
olamaz.</div>
```

```
<div *ngIf="personelForm.get('personel_email')!.hasError('pattern')">Geçerli bir e-mail adresi  
giriniz.</div>
```

```
</div>
```

```
<h6 class="me-auto">Personel Şifre</h6>
```

```
<input id="pro-mal" formControlName="personel_sifre" minlength="1" required class="border-0 mb-  
2 rounded-5 in" type="text">
```

```
<div *ngIf="personelForm.get('personel_sifre')!.invalid && (personelForm.get('personel_sifre')!.dirty  
|| personelForm.get('personel_sifre')!.touched)">
```

```
<div *ngIf="personelForm.get('personel_sifre')!.hasError('required')">Şifre alanı boş olamaz.</div>
```

```
<div *ngIf="personelForm.get('personel_sifre')!.hasError('minlength')">Şifre en az 6 karakter  
uzunluğunda olmalıdır.</div>
```

```
</div>
```

```
<div id="pro-kat">
```

```
<h6 class="me-auto">Rol</h6>
```

```
<select formControlName="personel_rol" required class="border-0 mb-2 rounded-5 in">
```

```
<option value="" disabled selected hidden>Seçiniz</option>
```

```
<ng-container *ngFor="let roller of roller; let i = index">
```

```
<option *ngIf="roller.ls !== 1&&roller.ls !== 3" [value]="roller.ls">
```

```
  {{ roller.lisans_adlari }}
```

```
</option>
```

```
</ng-container>
```

```
</select>
```

```
<div *ngIf="personelForm.get('personel_rol')!.invalid && (personelForm.get('personel_rol')!.dirty ||  
personelForm.get('personel_rol')!.touched)">
```

```
<div *ngIf="personelForm.get('personel_rol')!.hasError('required')">Rol seçimi zorunludur.</div>
```

```
</div>
```

```
</div>
```

```
<h6 class="me-auto">Personel İsim</h6>

<input id="pro-mal" formControlName="personel_isim" minlength="1" required class="border-0
mb-2 rounded-5 in" type="text">

<div *ngIf="personelForm.get('personel_isim')!.invalid &&
(personelForm.get('personel_isim')!.dirty || personelForm.get('personel_isim')!.touched)">

<div *ngIf="personelForm.get('personel_isim')!.hasError('required')">İsim alanı boş
olamaz.</div>

<div *ngIf="personelForm.get('personel_isim')!.hasError('pattern')">Sadece harfler
girilebilir.</div>

</di>

<h6 class="me-auto pt-3"></h6>

<button *ngIf="this.drm==0" (click)="insertyenipersonel()" class="btn mt-4 my-3 d-block
rounded-5 border border-2 border-secondary fs-6 w-100 btn-outline-secondary"
[disabled]="personelForm.invalid">

    Personel Ekle!

</button>

<button *ngIf="this.drm==1" (click)="uptadeTamamla()" class="btn mt-4 my-3 d-block rounded-
5 border border-2 border-secondary fs-6 w-100 btn-outline-secondary" [disabled]="personelForm.invalid">

    Personel Güncelle!

</button>

</div>

</div>

</div>

</form>
```

```
<div id="blocker" class="blocker d-none"></div>
```

-----  
**----islemci.component.html----**

```
<navbar></navbar>

<div class="w-100">

    <ul id="list" class="main-opt list-unstyled my-0 w-100 text-white">

        <li id="0" routerLink="/islemci" routerLinkActive="Sitem" [routerLinkActiveOptions]="{exact : true}"
class="list-item " >İstekler</li>
```



```
<li id="1" routerLink="/islemci/ekle" class="list-item" [routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >urunler ve ayrıntıları</li>

<li id="2" routerLink="/islemci/hazirlanan" class="list-item"[routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >hazirlanan Ürünler</li>

<li id="3" routerLink="/islemci/gecmis" class="list-item"[routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >Sipariş Geçmişi</li>

</ul>

</div>

<div class="font">

<div class="card-bg mb-4 px-2 position-relative mx-4 text-white ">

<div class="table-responsive max-overflow">

<table class="table table-borderless text-center text-white ">

<thead>

<tr>

<th class="text" scope="col">Ürün id</th>

<th class="text" scope="col">Ürün ad</th>

<th class="text" scope="col">Kategori</th>

<th class="text" scope="col">Marka</th>

<th class="text" scope="col">stok</th>

<th class="text" scope="col">onaylanabilir adet</th>

<th class="text" scope="col">siparis adeti</th>

<th class="text" scope="col">Fiyat</th>

<th class="text" scope="col">firma</th>

<th class="text" scope="col">Tarih</th>

<th class="text" scope="col">İşlem</th>

</tr>

</thead>

<tbody class="w-100 "><!--ngFor-->

<tr class="" *ngFor="let urun of islemcisepet">

<td class="text col text-break"scope="row">{{ urun.urun_ID }}</td>

<td class="text col-2 text-break">{{ urun.urun_ad }}</td>
```

```
<td class="text col text-break">{{urun.kategori_AD}}</td>

<td class="text col text-break">{{urun.marka_AD}}</td>

<td class="text col text-break">{{urun.urun_adet}}</td>

<td class="text col text-break">{{urun.iss}}</td>

<td class="text col text-break">{{urun.siparis_adeti}}</td>

<td class="text col text-break">{{urun.satis}}</td>

<td class="text col text-break">{{urun.Sirket_AD}}</td>

<td class="text col text-break">{{urun.siparis_tarih | date:'dd/MM/yyyy'}}</td>

<td class="text col">

    <button *ngIf="urun.urun_adet < urun.siparis_adeti" [disabled]="true" (click)="satinal(urun)"
id="{{urun.id + urun.firma + 1}}" type="button" class="btn btn-sm text-white w-100 succ">onayla</button>

    <button *ngIf="urun.urun_adet >= urun.siparis_adeti" [disabled]="false" (click)="satinal(urun)"
id="{{urun.id + urun.firma + 1}}" type="button" class="btn btn-sm text-white w-100 succ">onayla</button>

    <button id="{{urun.id+urun.firma}}" (click)="sil(urun)" type="button" class="btn btn-sm text-
white w-100 btn-danger">iptal et</button>

</td>

</tr>

</tbody>

</table>

</div>

</div>

</div>
```

---

#### ----ekle.component.html----

```
<navbar></navbar>

<div class="w-100">

  <ul id="list" class="main-opt list-unstyled my-0 w-100 text-white">

    <li id="0" routerLink="/islemci" routerLinkActive="Sitem" [routerLinkActiveOptions]="{exact : true}"
class="list-item " >İstekler</li>

    <li id="1" routerLink="/islemci/ekle" class="list-item" [routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >urunler ve ayrıntıları</li>

    <li id="2" routerLink="/islemci/hazirlanan" class="list-item"[routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >hazirlanan Ürünler</li>

    <li id="3" routerLink="/islemci/gecmis" class="list-item"[routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >Sipariş Geçmişi</li>

  </ul>

</div>

<app-product></app-product>
```

#### ----product.component.html----

```
<div class="font">

  <div class="card-bg stock-cont text-center my-3 px-md-2 px-0 mx-0 mx-md-4 text-white ">

    <div class="py-3 st-1 ps-3">

      Toplam Ürün Çeşit Sayısı

      <br>

      <h3 class="my-5 d-inline-block yen rounded-5 px-5 py-1">{{ veri1.length }}</h3>

    </div>

    <div class="st-2 ">

      <h3 class="position-relative w-75 text-center text-white rounded-5 prod my-3 d-inline-block px-5 py-2">Ürünler</h3><br>

      <h5 class="d-inline-block">Ürün Ara</h5><br>

      <input type="text" (change)="ara($event)" class="in w-75 rounded-5"><br>

      <br>

      <button (click)="alphabetA()" class="btn btn-primary my-3 me-4">A-Z</button>

      <button class="btn yen my-3 text-white me-4" (click)="pro_ac()">Ürün Ekle</button>
```

<button class="btn succ my-3 text-white me-4" (click)="mark\_ac()">Marka Ekle</button>

<button class="btn succ my-3 text-white me-4" (click)="kat\_ac()">Kategori Ekle</button>

<button (click)="alphabetZ()" class="btn btn-secondary my-3">Z-A</button>

</div>

<div class="py-3 st-3 pe-3">

Toplam Kategori Sayısı

<br>

<h3 class="my-5 d-inline-block sat rounded-5 px-5 py-1">{{ kategoriler.length }}</h3>

</div>

</div>

<div class="font">

<div class="card-bg mb-4 px-md-2 px-0 mx-0 mx-md-4 text-white ">

<div class="table-responsive max-overflow">

<table class="table table-borderless text-center text-white">

<thead>

<tr>

<th class="text" scope="col">Ürün id</th>

<th class="text" scope="col">Ürün ad</th>

<th class="text" scope="col">Kategori</th>

<th class="text" scope="col">Marka</th>

<th class="text" scope="col">Ayrıntılar</th>

<th class="text" scope="col">Stok</th>

<th class="text" scope="col">satılabilir stok</th>

<th class="text" scope="col">Fiyat</th>

<th class="text" scope="col">Maliyet</th>

</tr>

</thead>

<tbody class="w-100">

<tr class="" \*ngFor="let urun of veri1copy">

```

<td *ngIf="urun.kategori_ID>=1" class="text col text-break"scope="row">{{urun.urun_ID}}</td>
<td *ngIf="urun.kategori_ID>=1" class="text col-2 text-break">{{urun.urun_ad}}</td>
<td *ngIf="urun.kategori_ID>=1" class="text col text-break">{{urun.kategori_AD}}</td>
<td *ngIf="urun.kategori_ID>=1" class="text col text-break">{{urun.marka_AD}}</td>
<td *ngIf="urun.kategori_ID>=1" class="text col">
    <button id="edit-pro" type="button"(click)="this.drm=1" (click)="acikla(urun.aciklama)"
class="btn btn-sm h-100 text-white edit">Ayrıntılar</button>

```

```

</td>
<td *ngIf="urun.kategori_ID>=1" class="text col text-break">{{urun.urun_adet}}</td>
<td *ngIf="urun.kategori_ID>=1" class="text col text-break">{{urun.iss}}</td>
<td *ngIf="urun.kategori_ID>=1" class="text col text-break">{{urun.satis}}</td>
<td *ngIf="urun.kategori_ID>=1" class="text col text-break">{{urun.maliyet}}</td>
<td *ngIf="urun.kategori_ID>=1" class="text text-break">
    <button type="button" (click)="updateUrun(urun)" class="btn btn-sm text-white w-100 btn-
warning">Düzenle</button><br>
    <button type="button" (click)="deleteUrun(urun.urun_ID)" class="btn btn-sm text-white w-100
btn-danger">Sil</button>

```

```

</td>
</tr>
</tbody>
</table>
</div>

</div>

</div>

```

```

<form [formGroup]="productForm" >
<div id="pro-add" class="d-none login p-3 my-auto border-0 mx-auto rounded-5">
<i id="cross" (click)="pro_kapat()" class="bi bi-x-lg cross"></i>
<div id="pro-form" class="log py-5 p-25 rounded-5">
<div id="pro-frm">

```

```
<h6 class="me-auto">Ürün Adı</h6>

<input id="pro-id" formControlName="productName" required maxlength="50" minlength="3"
class="border-0 mb-2 rounded-5 in" type="text">

<div *ngIf="productForm.get('productName')!.invalid && (productForm.get('productName')!.dirty ||
productForm.get('productName')!.touched)">

  <div *ngIf="productForm.get('productName')!.hasError('required')">Kategori alanı seçiniz.</div>

  <div *ngIf="productForm.get('productName')!.hasError('minlength')">Minimum karakter sayısını
geçin.</div>

</div>
```

```
<h6 class="me-auto">Kategori</h6>

<select id="pro-kat" formControlName="category" required class="border-0 mb-2 rounded-5 in">

  <ng-container *ngFor="let kat of kategoriler">

    <option *ngIf="kat.kategori_ID>=1" [value]="kat.kategori_ID">{{ kat.kategori_AD }}</option>

  </ng-container>

</select>
```

```
<div *ngIf="productForm.get('category')!.invalid && (productForm.get('category')!.dirty ||
productForm.get('category')!.touched)">

  <div *ngIf="productForm.get('category')!.hasError('required')">Kategori alanı seçiniz.</div>

</div>
```

```
<h6 class="me-auto">Marka</h6>

<select id="pro-mark" formControlName="brand" required class="border-0 mb-2 rounded-5 in">

  <option *ngFor="let mak of markalar" value={{ mak.marka_ID }}>{{ mak.marka_AD }}</option>

</select>

<div *ngIf="productForm.get('brand')!.invalid && (productForm.get('brand')!.dirty ||
productForm.get('brand')!.touched)">

  <div *ngIf="productForm.get('brand')!.hasError('required')">Kategori alanı seçiniz.</div>

</div>
```

<h6 class="me-auto">Maliyet</h6>

<input id="pro-mal" formControlName="cost" minlength="1" required class="border-0 mb-2 rounded-5 in" type="text">

<div \*ngIf="productForm.get('cost')!.invalid && (productForm.get('cost')!.dirty || productForm.get('cost')!.touched)">

<div \*ngIf="productForm.get('cost')!.hasError('required')">Bu alan boş bırakılamaz.</div>

<div \*ngIf="productForm.get('cost')!.hasError('pattern')">Bu alana sadece sayı girilmelidir.</div>

</div>

<h6 class="me-auto">Satış</h6>

<input id="pro-sat" formControlName="sale" minlength="1" required class="border-0 mb-2 rounded-5 in" type="text">

<div \*ngIf="productForm.get('sale')!.invalid && (productForm.get('sale')!.dirty || productForm.get('sale')!.touched)">

<div \*ngIf="productForm.get('sale')!.hasError('required')">Bu alan boş bırakılamaz.</div>

<div \*ngIf="productForm.get('sale')!.hasError('pattern')">Bu alana sadece sayı girilmelidir.</div>

</div>

<h6 class="me-auto">Ürün Açıklaması</h6>

<textarea id="pro-acikla" placeholder="sadece 100 karakter" formControlName="exp" required maxlength="100" minlength="3" class="border-0 mb-2 rounded-5 in" ></textarea>

<div \*ngIf="productForm.get('exp')!.invalid && (productForm.get('exp')!.dirty || productForm.get('exp')!.touched)">

<div \*ngIf="productForm.get('exp')!.hasError('required')">Kategori alanı seçiniz.</div>

<div \*ngIf="productForm.get('exp')!.hasError('minlength')">Minimum karakter sayısını geçin.</div>

</div>

<h6 class="me-auto pt-3"></h6>

<button \*ngIf="this.drm==0" (click)="insertYeniurun()" class="btn mt-4 my-3 d-block rounded-5 border border-2 border-secondary fs-6 w-100 btn-outline-secondary" [disabled]="productForm.invalid">

Ürünü Ekle!

</button>

<button \*ngIf="this.drm==1" (click)="uptadeTamamla()" class="btn mt-4 my-3 d-block rounded-5 border border-2 border-secondary fs-6 w-100 btn-outline-secondary" [disabled]="productForm.invalid">

Ürünü Güncelle!

</button>

</div>

</div>

</div>

</form>

<form [formGroup]="categoryForm" (ngSubmit)="insertyenikategori(">

<div id="kat-add" class="d-none login p-3 my-auto border-0 mx-auto rounded-5">

<i id="cross" (click)="kat\_kapat()" class="bi bi-x-lg cross"></i>

<div id="pro-form" class="log py-5 p-25 rounded-5">

<div id="pro-frm">

<h6 class="me-auto">Kategori Adı</h6>

<input id="pro-id" formControlName="katName" required maxlength="50" minlength="3" class="border-0 mb-2 rounded-5 in" type="text">

<div \*ngIf="categoryForm.get('katName')!.invalid && (categoryForm.get('katName')!.dirty || categoryForm.get('katName')!.touched)">

<div \*ngIf="categoryForm.get('katName')!.hasError('required')>Boş geçilemez.</div>

<div \*ngIf="categoryForm.get('katName')!.hasError('minlength')>Minimum karakter sayısını geçin.</div>

</div>

<button class="btn mt-4 my-3 d-block rounded-5 border border-2 border-secondary fs-6 w-100 btn-outline-secondary" [disabled]="categoryForm.invalid">

Kategori Ekle!

</button>

</div></div></div></form>



```

<form [formGroup]="markForm" (ngSubmit)="insertyenimarka()">

  <div id="mark-add" class="d-none login p-3 my-auto border-0 mx-auto rounded-5">

    <i id="cross" (click)="mark_kapat()" class="bi bi-x-lg cross"></i>

    <div id="pro-form" class="log py-5 p-25 rounded-5">

      <div id="pro-frm">

        <h6 class="me-auto">Marka Adı</h6>

        <input id="pro-id" formControlName="markName" required maxlength="50" minlength="3"
class="border-0 mb-2 rounded-5 in" type="text">

        <div *ngIf="markForm.get('markName')!.invalid && (markForm.get('markName')!.dirty ||
markForm.get('markName')!.touched)">

          <div *ngIf="markForm.get('markName')!.hasError('required')">Kategori alanı seçiniz.</div>

          <div *ngIf="markForm.get('markName')!.hasError('minlength')">Minimum karakter sayısını
geçin.</div>

        </div>

        <button class="btn mt-4 my-3 d-block rounded-5 border border-2 border-secondary fs-6 w-100 btn-
outline-secondary" [disabled]="markForm.invalid">

          Marka Ekle!

        </button>

      </div></div></div></form>

```

```

<div id="ac_ekran" class="d-none login p-3 my-auto border-0 mx-auto rounded-5">

  <i id="cross" (click)="aciklama_kapat()" class="bi bi-x-lg cross"></i>

  <div id="aciklama-form" class="log py-5 p-25 rounded-5">

    <div id="aciklama"></div>

  </div></div></div>

```

```

<div id="blocker" class="blocker d-none"></div>

```

-----

----gecmis.component.html----

```

<navbar></navbar>

```

```
<div class="w-100">

  <ul id="list" class="main-opt list-unstyled my-0 w-100 text-white">

    <li id="0" routerLink="/islemci" routerLinkActive="Sitem" [routerLinkActiveOptions]={exact : true}"
class="list-item " >İstekler</li>

    <li id="1" routerLink="/islemci/ekle" class="list-item" [routerLinkActiveOptions]={exact : true}"
routerLinkActive="Sitem" >urunler ve ayrıntıları</li>

    <li id="2" routerLink="/islemci/hazirlanan" class="list-item"[routerLinkActiveOptions]={exact : true}"
routerLinkActive="Sitem" >hazirlanan Ürünler</li>

    <li id="3" routerLink="/islemci/gecmis" class="list-item"[routerLinkActiveOptions]={exact : true}"
routerLinkActive="Sitem" >Sipariş Geçmişi</li>

  </ul>

</div>

<app-gecmis></app-gecmis>
```

#### ----hazirlanan.component.html----

```
<navbar></navbar>

<div class="w-100">

  <ul id="list" class="main-opt list-unstyled my-0 w-100 text-white">

    <li id="0" routerLink="/islemci" routerLinkActive="Sitem" [routerLinkActiveOptions]={exact : true}"
class="list-item " >İstekler</li>

    <li id="1" routerLink="/islemci/ekle" class="list-item" [routerLinkActiveOptions]={exact : true}"
routerLinkActive="Sitem" >urunler ve ayrıntıları</li>

    <li id="2" routerLink="/islemci/hazirlanan" class="list-item"[routerLinkActiveOptions]={exact : true}"
routerLinkActive="Sitem" >hazirlanan Ürünler</li>

    <li id="3" routerLink="/islemci/gecmis" class="list-item"[routerLinkActiveOptions]={exact : true}"
routerLinkActive="Sitem" >Sipariş Geçmişi</li>

  </ul>

</div>

<app-sell-calendar (DateEvent)="Date($event)" (selectedDateEvent)="handleSelectedDate($event)" ></app-
sell-calendar>

<div class="font">

  <div class="card-bg mb-4 px-2 px-md-2 mx-0 px-0 my-3 mx-md-4 mx-4 text-white ">

    <div class="table-responsive max-overflow">

      <table class="table table-borderless text-center text-white ">
```

```

<thead>

<tr>

<th class="text" scope="col">Ürün id</th>

<th class="text" scope="col">Ürün ad</th>

<th class="text" scope="col">Kategori</th>

<th class="text" scope="col">Marka</th>

<th class="text" scope="col">Adet</th>

<th class="text" scope="col">Firma</th>

<th class="text" scope="col">Tutar</th>

<th class="text" scope="col">Sipariş durum</th>

<th class="text" scope="col">İşlem</th>

</tr>

</thead>

<tbody class="w-100"><!--ngFor-->

<tr class="" *ngFor="let urun of veriler">

<td class="text col text-break" scope="row">{{ urun.urun_ID }}</td>

<td class="text col-2 text-break">{{ urun.urun_ad }}</td>

<td class="text col text-break">{{ urun.kategori_AD }}</td>

<td class="text col text-break">{{ urun.marka_AD }}</td>

<td class="text col text-break">{{ urun.siparis_adeti }}</td>

<td class="text col text-break">AH firması</td>

<td class="text col text-break">{{ (urun.urun_adet*urun.satis) }} TL</td>

<td class="text col text-break">{{ urun.durum }}</td>

<td class="text col">

<button id={{ urun.id+urun.firma }} (click)="sil(urun)" type="button" class="btn btn-sm text-white
h-100 btn-danger">siparis iptal</button>

</td>

</tr>

</tbody>

```

</table>

</div>

</div>

</div>

## ----sell-calendar.component.html----

<head>

<script type="modal" src="../../assets/js/date.js"></script>

</head>

<div class="py-4 font px-0 call-design">

<div id="callendar" class="card-bg call-area w-100 d-inline-block position-relative text-center text-white ">

<h3 class="py-4">Şipariş Takvimi</h3>

<div class="w-100 d-flex align-items-center justify-content-center mt-3 mb-5">

<div \*ngIf="this.bugun.getMonth()!==this.startDate.getMonth()" class="d-inline-block m-0 p-0" id="geri" (click)="this.prev()"><i class="d-inline-block carousel-control-prev-icon "></i></div>

<h3 class="d-inline-block w-50 my-0">{{ay}}</h3>

<div class="d-inline-block m-0 p-0" id="ileri" (click)="this.next()"><i class="d-inline-block carousel-control-next-icon "></i></div>

</div>

<table class="table cal table-bordered text-center text-white ">

<thead>

<tr>

<th scope="col text">Pzt</th>

<th scope="col text">Salı</th>

<th scope="col text">Çar</th>

<th scope="col text">Per</th>

<th scope="col text">Cum</th>

<th scope="col text">Cmt</th>

<th scope="col text">Pzr</th>

</tr>

</thead>

```

<tbody id="tbod" class="">

  <tr *ngFor="let hafta of calendar">

    <ng-container *ngFor="let sut of hafta">

      <td class="text">

        <div (click)="selectDate(sut,startDate.getMonth(),startDate.getFullYear())" *ngIf="sut !== null; else
empty"[ngClass]='{cal-item ': true

          , 'smaller-day': sut < bugun.getDate() && bugun.getMonth() === startDate.getMonth() &&
bugun.getFullYear() === startDate.getFullYear()

          , 'small-sip-day': isSmallSipDay(sut)

          , 'sip-day': isSipDay(sut)}">{{ sut }}</div>

        <ng-template #empty></ng-template>

      </td>

    </ng-container>

  </tr>

</tbody>

  <button (click)="tumSip()" class="position-absolute bottom-0 end-0 btn px-4 btn-secondary text-white
rounded-5">Tüm Siparişler</button>

</table>

</div>
</div>

```

#### ----depo.component.html----

```

<navbar></navbar>

<div>

  <ul id="list" class="main-opt2 list-unstyled my-0 w-100 text-white">

    <li id="0" routerLink="/depo" routerLinkActive="Sitem" [routerLinkActiveOptions]="{exact : true}"
class="list-item" >depo</li>

    <li id="1" routerLink="/depo/kayit" class="list-item " [routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >kayit</li>

  </ul>

```

```

<div class="font">

<div class="card-bg mb-4 px-2 px-md-2 mx-0 px-0 my-3 mx-md-4 mx-4 text-white ">

<div class="table-responsive max-overflow">

<table class="table table-borderless text-center text-white ">

<thead>

<tr>

<th class="text" scope="col">Ürün id</th>

<th class="text" scope="col">Ürün ad</th>

<th class="text" scope="col">Kategori</th>

<th class="text" scope="col">Marka</th>

<th class="text" scope="col">Adet</th>

<th class="text" scope="col">Firma</th>

<th class="text" scope="col">Tutar</th>

<th class="text" scope="col">Sipariş durum</th>

<th class="text" scope="col">İşlem</th>

</tr>

</thead>

<tbody class="w-100"><!--ngFor-->

<tr class="" *ngFor="let urun of veriler">

<td class="text col text-break"scope="row">{{ urun.urun_ID }}</td>

<td class="text col-2 text-break">{{ urun.urun_ad }}</td>

<td class="text col text-break">{{ urun.kategori_AD }}</td>

<td class="text col text-break">{{ urun.marka_AD }}</td>

<td class="text col text-break">{{ urun.siparis_adeti }}</td>

<td class="text col text-break">Magaza 1</td>

<td class="text col text-break">{{ (urun.urun_adet*urun.satis) }} TL</td>

<td class="text col text-break "><span class="bg-primary p-3 rounded-5">{{ urun.durum }}</span></td>

<td class="text col">

```

```
<button id={ {urun.id+urun.firma} } (click)="sil(urun)" type="button" class="btn btn-sm text-white w-100 h-50 btn-danger">siparis iptal</button>
```

```
<button id={ {urun.id+urun.firma} } (click)="bitir(urun)" type="button" class="btn btn-sm text-white w-100 h-50 succ">Tamamlandı</button>
```

```
</td>
```

```
</tr>
```

```
</tbody>
```

```
</table>
```

```
</div>
```

```
</div>
```

```
</div>
```

#### ----kayit.component.html----

```
<navbar></navbar>
```

```
<div>
```

```
<ul id="list" class="main-opt2 list-unstyled my-0 w-100 text-white">
```

```
<li id="0" routerLink="/depo" routerLinkActive="Sitem" [routerLinkActiveOptions]="{exact : true}" class="list-item" >depo</li>
```

```
<li id="1" routerLink="/depo/kayit" class="list-item " [routerLinkActiveOptions]="{exact : true}" routerLinkActive="Sitem" >kayı</li>
```

```
</ul>
```

```
</div>
```

```
<div class="font">
```

```
<div class="card-bg stock-cont text-center my-3 px-md-2 px-0 mx-0 mx-md-4 text-white ">
```

```
<div class="py-3 st-1 ps-3">
```

Toplam Ürün Çeşit Sayısı

```
<br>

<h3 class="my-5 d-inline-block yen rounded-5 px-5 py-1">{{ veri1.length }}</h3>

</div>

<div class="st-2 ">

<h3 class="position-relative w-75 text-center text-white rounded-5 prod my-3 d-inline-block px-5 py-2">Ürünler</h3><br>

<h5 class="d-inline-block">Ürün Ara</h5><br>

<input type="text" (change)="ara($event)" class="in w-75 rounded-5"><br>

<button (click)="alphabetA()" class="btn btn-primary my-3 me-4">A-Z</button>

<button (click)="alphabetZ()" class="btn btn-secondary my-3">Z-A</button>

</div>

<div class="py-3 st-3 pe-3">

  Toplam Kategori Sayısı

  <br>

  <h3 class="my-5 d-inline-block sat rounded-5 px-5 py-1">{{ kategoriler.length }}</h3>

</div>

</div>

<div class="card-bg mb-4 px-2 mx-4 text-white ">

<div class="table-responsive max-overflow" >

  <table class="table table-borderless text-center text-white ">

    <thead>

      <tr>

        <th class="text" scope="col">Ürün id</th>

        <th class="text" scope="col">Ürün ad</th>

        <th class="text" scope="col">Kategori</th>

        <th class="text" scope="col">Marka</th>

        <th class="text" scope="col">Ayrıntılar</th>

        <th class="text" scope="col">Stok</th>

        <th class="text" scope="col">Firma</th>

      </tr>

    </thead>

  </table>

</div>

</div>
```



```

        <th class="text" scope="col">Fiyat</th>

        <th class="text" scope="col">İşlem</th>

    </tr>

</thead>

<tbody class="w-100"><!--ngFor-->

    <tr class="" *ngFor="let urun of veri1copy">

        <td *ngIf="urun.kategori_ID>=1" id="urunid" class="text col text-
break" scope="row">{{ urun.urun_ID }}</td>

        <td *ngIf="urun.kategori_ID>=1" class="text col-2 text-break">{{ urun.urun_ad }}</td>

        <td *ngIf="urun.kategori_ID>=1" class="text col text-break">{{ urun.kategori_AD }}</td>

        <td *ngIf="urun.kategori_ID>=1" class="text col text-break">{{ urun.marka_AD }}</td>

        <td *ngIf="urun.kategori_ID>=1" class="text col">

            <button id="edit-pro" type="button"(click)="this.drm=1" (click)="accc(urun)" class="btn btn-sm h-
100 text-white edit">Ayrıntılar</button>

        </td>

        <td *ngIf="urun.kategori_ID>=1" class="text col text-break">{{ urun.iss }}</td>

        <td *ngIf="urun.kategori_ID>=1" class="text col text-break">AH firması</td>

        <td *ngIf="urun.kategori_ID>=1" class="text col text-break">{{ urun.satis }}</td>

        <td *ngIf="urun.kategori_ID>=1" class="text col">

            <!--<button id={{ urun.id+urun.firma }} (click)="satinal(urun,$event)" type="button" class="btn btn-
sm text-white h-100 btn-secondary">Sepete Ekle</button>-->

            <button id={{ urun.id+urun.firma }} (click)="adetekle(urun,adet)" type="button" class="btn btn-sm
text-white h-100 btn-secondary">adet ekle</button>

        </td>

    </tr>

</tbody>

</table>

</div>

</div>

```

```
<div id="panel" class="d-none login p-3 border-0 mx-auto rounded-5 ">

  <i id="cross" (click)="kapt()" class="bi bi-x-lg cross"></i>

  <div class="log p-25 p-3 rounded-5">

    <div id="logs">

      <div>

    </div>

  </div>

</div>

<div id="aciklama" class="d-none"></div>

<div id="satal" class="d-none">

  <h6 class="me-auto pt-3 ">Kaç Adet</h6>

  <input id="adet" #myInput class="border-0 rounded-5 in" type="text">

  <button id="confirm" (click)="tamamla(myInput.value)" class="btn mt-4 my-3 d-block rounded-5 border
border-2 border-secondary fs-6 w-100 btn-outline-secondary" >

    Tamamla!

</button>

</div>

</div> </div></div></div>

</div>
```

#### ----magaza.component.html----

```
<navbar></navbar>

<div>

  <ul id="list" class="main-opt list-unstyled my-0 w-100 text-white">

    <li id="0" routerLink="/magaza" routerLinkActive="Sitem" [routerLinkActiveOptions]="{exact : true}"
class="list-item " >ürünler</li>

    <li id="1" routerLink="/magaza/sepet" class="list-item" [routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >sepetim</li>

    <li id="2" routerLink="/magaza/siparis" class="list-item"[routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >siparişlerim</li>

    <li id="3" routerLink="/magaza/gecmis" class="list-item"[routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >Sipariş geçmişi</li>

  </ul>

</div>
```

```
<div class="font">

  <div class="card-bg stock-cont text-center my-3 px-md-2 px-0 mx-0 mx-md-4 text-white ">

    <div class="py-3 st-1 ps-3">

      Toplam Ürün Çeşit Sayısı

      <br>

      <h3 class="my-5 d-inline-block yen rounded-5 px-5 py-1">{{ veri1.length }}</h3>

    </div>

    <div class="st-2 ">

      <h3 class="position-relative w-75 text-center text-white rounded-5 prod my-3 d-inline-block px-5 py-2">Ürünler</h3><br>

      <h5 class="d-inline-block">Ürün Ara</h5><br>

      <input type="text" (change)="ara($event)" class="in w-75 rounded-5"><br>

      <button (click)="alphabetA()" class="btn btn-primary my-3 me-4">A-Z</button>

      <button (click)="alphabetZ()" class="btn btn-secondary my-3">Z-A</button>

    </div>

    <div class="py-3 st-3 pe-3">

      Toplam Kategori Sayısı

      <br>

      <h3 class="my-5 d-inline-block sat rounded-5 px-5 py-1">{{ kategoriler.length }}</h3>

    </div>

  </div>

<div class="card-bg mb-4 px-2 mx-4 text-white ">

<div class="table-responsive max-overflow" >

  <table class="table table-borderless text-center text-white ">

    <thead>

      <tr>

        <th class="text" scope="col">Ürün id</th>

        <th class="text" scope="col">Ürün ad</th>

        <th class="text" scope="col">Kategori</th>

        <th class="text" scope="col">Marka</th>

        <th class="text" scope="col">Ayrıntılar</th>

      </tr>

    </thead>

    <tbody>

      <tr>

        <td>1</td>

        <td>Ürün 1</td>

        <td>Kategori 1</td>

        <td>Marka 1</td>

        <td>Ayrıntılar 1</td>

      </tr>

      <tr>

        <td>2</td>

        <td>Ürün 2</td>

        <td>Kategori 2</td>

        <td>Marka 2</td>

        <td>Ayrıntılar 2</td>

      </tr>

      <tr>

        <td>3</td>

        <td>Ürün 3</td>

        <td>Kategori 3</td>

        <td>Marka 3</td>

        <td>Ayrıntılar 3</td>

      </tr>

      <tr>

        <td>4</td>

        <td>Ürün 4</td>

        <td>Kategori 4</td>

        <td>Marka 4</td>

        <td>Ayrıntılar 4</td>

      </tr>

      <tr>

        <td>5</td>

        <td>Ürün 5</td>

        <td>Kategori 5</td>

        <td>Marka 5</td>

        <td>Ayrıntılar 5</td>

      </tr>

      <tr>

        <td>6</td>

        <td>Ürün 6</td>

        <td>Kategori 6</td>

        <td>Marka 6</td>

        <td>Ayrıntılar 6</td>

      </tr>

      <tr>

        <td>7</td>

        <td>Ürün 7</td>

        <td>Kategori 7</td>

        <td>Marka 7</td>

        <td>Ayrıntılar 7</td>

      </tr>

      <tr>

        <td>8</td>

        <td>Ürün 8</td>

        <td>Kategori 8</td>

        <td>Marka 8</td>

        <td>Ayrıntılar 8</td>

      </tr>

      <tr>

        <td>9</td>

        <td>Ürün 9</td>

        <td>Kategori 9</td>

        <td>Marka 9</td>

        <td>Ayrıntılar 9</td>

      </tr>

      <tr>

        <td>10</td>

        <td>Ürün 10</td>

        <td>Kategori 10</td>

        <td>Marka 10</td>

        <td>Ayrıntılar 10</td>

      </tr>

    </tbody>

  </table>

</div>

</div>
```

```

<th class="text" scope="col">Stok</th>

<th class="text" scope="col">Firma</th>

<th class="text" scope="col">Fiyat</th>

<th class="text" scope="col">İşlem</th>

</tr>

</thead>

<tbody class="w-100"><!--ngFor-->

<tr class="" *ngFor="let urun of veri1copy">

  <td *ngIf="urun.kategori_ID>=1" class="text col text-break" scope="row">{{ urun.urun_ID }}</td>

  <td *ngIf="urun.kategori_ID>=1" class="text col-2 text-break">{{ urun.urun_ad }}</td>

  <td *ngIf="urun.kategori_ID>=1" class="text col text-break">{{ urun.kategori_AD }}</td>

  <td *ngIf="urun.kategori_ID>=1" class="text col text-break">{{ urun.marka_AD }}</td>

  <td *ngIf="urun.kategori_ID>=1" class="text col">

    <button id="edit-pro" type="button"(click)="this.drm=1" (click)="acce(urun)" class="btn btn-sm h-100 text-white edit">Ayrıntılar</button>

  </td>

  <td *ngIf="urun.kategori_ID>=1" class="text col text-break">{{ urun.iss }}</td>

  <td *ngIf="urun.kategori_ID>=1" class="text col text-break">AH firması</td>

  <td *ngIf="urun.kategori_ID>=1" class="text col text-break">{{ urun.satis }}</td>

  <td *ngIf="urun.kategori_ID>=1" class="text col">

    <button *ngIf="urun.iss>0" id={{ urun.id+urun.firma }} (click)="satinal(urun,$event)" type="button" class="btn btn-sm text-white w-100 h-100 btn-secondary">Sepete Ekle</button>

    <button *ngIf="urun.iss<=0" id={{ urun.id+urun.firma }} (click)="satinal(urun,$event)" disabled="true" type="button" class="btn w-100 btn-sm text-white h-100 btn-danger">Stokta yok</button>

  </td>

</tr>

</tbody>

</table>

</div>

</div>

<div id="panel" class="d-none login p-3 border-0 mx-auto rounded-5 ">

```

```
<i id="cross" (click)="kapt()" class="bi bi-x-lg cross"></i>

<div class="log p-25 p-3 rounded-5">

  <div id="logs">

    <div>

<div id="aciklama" class="d-none"></div>

<div id="satal" class="d-none">

  <h6 class="me-auto pt-3 ">Kaç Adet</h6>

  <input id="adet" class="border-0 rounded-5 in" type="text">

  <button id="confirm" (click)="tamamla()" class="btn mt-4 my-3 d-block rounded-5 border border-2 border-
secondary fs-6 w-100 btn-outline-secondary" >

    Tamamla!

</button>

</div>

</div>

</div></div></div>
```

</div>

#### ----gecmis.html----

```
<div *ngIf="router.url === '/magaza/gecmis'">

<navbar></navbar>

<div>

  <ul id="list" class="main-opt list-unstyled my-0 w-100 text-white">

    <li id="0" routerLink="/magaza" routerLinkActive="Sitem" [routerLinkActiveOptions]="{exact : true}"
class="list-item " >ürünler</li>

    <li id="1" routerLink="/magaza/sepet" class="list-item" [routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >sepetim</li>

    <li id="2" routerLink="/magaza/siparis" class="list-item"[routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >siparişlerim</li>

    <li id="3" routerLink="/magaza/gecmis" class="list-item"[routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >Sipariş geçmişi</li>

  </ul>

</div>

</div>
```

```
<div class="font">
```

```
<div class="card-bg mb-4 px-2 px-md-2 mx-0 px-0 my-3 mx-md-4 mx-4 text-white ">
```

```
<div class="table-responsive max-overflow">
```

```
<table class="table table-borderless text-center text-white ">
```

```
<thead>
```

```
<tr>
```

```
<th class="text" scope="col">Ürün id</th>
```

```
<th class="text" scope="col">Ürün ad</th>
```

```
<th class="text" scope="col">Kategori</th>
```

```
<th class="text" scope="col">Marka</th>
```

```
<th class="text" scope="col">Adet</th>
```

```
<th class="text" scope="col">Firma</th>
```

```
<th class="text" scope="col">Tutar</th>
```

```
<th class="text" scope="col">Sipariş durum</th>
```

```
<th class="text" scope="col">Sipariş tarihi</th>
```

```
<th class="text" scope="col">Teslim tarihi</th>
```

```
</tr>
```

```
</thead>
```

```
<tbody class="w-100"><!--ngFor-->
```

```
<tr class="" *ngFor="let urun of gecmis">
```

```
<td class="text col text-break" scope="row">{{ urun.urun_ID }}</td>
```

```
<td class="text col-2 text-break">{{ urun.urun_ad }}</td>
```

```
<td class="text col text-break">{{ urun.kategori_AD }}</td>
```

```
<td class="text col text-break">{{ urun.marka_AD }}</td>
```

```
<td class="text col text-break">{{ urun.adet }}</td>
```

```
<td class="text col text-break">AH firması</td>
```

```
<td class="text col text-break">{{ (urun.adet*urun.satis) }} TL</td>
```

```
<td *ngIf="urun.durum=='sipariş iptal'" class="text col kırmızı text-break">{{ urun.durum }}</td>
```

```
<td *ngIf="urun.durum=='sipariş gönderildi'" class="text col yeşil text-break">{{ urun.durum }}</td>
```

```

        <td class="text col text-break">{{ urun.siparis_tarih | date:'dd/MM/yyyy' }}</td>

        <td class="text col text-break">{{ urun.teslim_tarihi | date:'dd/MM/yyyy' }}</td>
    </tr>

</tbody>

</table>

</div>

</div>

</div>

```

#### ---hata.component.html---

```

<br>

<a href="/" class="bg-primary position-absolute top-0 start-50 translate-middle-x my-3 p-2 px-5 text-decoration-none text-white rounded-5">Ana Sayfa</a>

```

#### ----sepet.component.html----

```

<navbar></navbar>

<div>

    <ul id="list" class="main-opt list-unstyled my-0 w-100 text-white">

        <li id="0" routerLink="/magaza" routerLinkActive="Sitem" [routerLinkActiveOptions]="{exact : true}"
class="list-item ">ürünler</li>

        <li id="1" routerLink="/magaza/sepet" class="list-item" [routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >sepetim</li>

        <li id="2" routerLink="/magaza/siparis" class="list-item"[routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >siparişlerim</li>

        <li id="3" routerLink="/magaza/gecmis" class="list-item"[routerLinkActiveOptions]="{exact : true}"
routerLinkActive="Sitem" >Sipariş geçmişi</li>

    </ul>

</div>

<div class="font">

    <div class="card-bg mb-4 px-2 mx-4 text-white ">

```

```

<div class="table-responsive max-overflow">

  <table class="table table-borderless text-center text-white ">

    <thead>

      <tr>

        <th class="text" scope="col">Ürün id</th>

        <th class="text" scope="col">Ürün ad</th>

        <th class="text" scope="col">Kategori</th>

        <th class="text" scope="col">Marka</th>

        <th class="text" scope="col">stok</th>

        <th class="text" scope="col">siparis adeti</th>

        <th class="text" scope="col">Firma</th>

        <th class="text" scope="col">Fiyat</th>

        <th class="text" scope="col">İşlem</th>

      </tr>

    </thead>

    <tbody class="w-100"><!--ngFor-->

      <tr class="" *ngFor="let urun of sepet">

        <td class="text col text-break"scope="row">{{ urun.urun_ID }}</td>

        <td class="text col-2 text-break">{{ urun.urun_ad }}</td>

        <td class="text col text-break">{{ urun.kategori_AD }}</td>

        <td class="text col text-break">{{ urun.marka_AD }}</td>

        <td class="text col text-break">{{ urun.iss }}</td>

        <td class="text col text-break"><input min="1" max={{ urun.iss }} (change)="kontrol($event,urun)"
type="number" value={{ urun.siparis_adeti }}></td>

        <td class="text col text-break">AH firması</td>

        <td class="text col text-break">{{ urun.satis }}</td>

        <td class="text col">

          <button *ngIf="urun.urun_adet < urun.siparis_adeti" [disabled]="true" (click)="satinal(urun)"
id="{{ urun.id + urun.firma + 1 }}" type="button" class="btn btn-sm text-white h-100 succ">satın al</button>

          <button *ngIf="urun.urun_adet >= urun.siparis_adeti" [disabled]="false" (click)="satinal(urun)"
id="{{ urun.id + urun.firma + 1 }}" type="button" class="btn btn-sm text-white h-100 succ">satın al</button>

```



```
        <button id={ {urun.id+urun.firma} } (click)="sil(urun)" type="button" class="btn btn-sm text-white h-100 btn-danger">iptal et</button>
```

```
    </td>
```

```
</tr>
```

```
        <button (click)="hepsiniAl()" type="button" class="btn btn-sm text-white h-100 btn-secondary">hepsini satın al</button>
```

```
</tbody>
```

```
</table>
```

```
</div>
```

```
</div> </div>
```

#### ----siparisler.component.html----

```
<navbar></navbar>
```

```
<div>
```

```
    <ul id="list" class="main-opt list-unstyled my-0 w-100 text-white">
```

```
        <li id="0" routerLink="/magaza" routerLinkActive="Sitem" [routerLinkActiveOptions]="{exact : true}" class="list-item " >ürünler</li>
```

```
        <li id="1" routerLink="/magaza/sepet" class="list-item" [routerLinkActiveOptions]="{exact : true}" routerLinkActive="Sitem" >sepetim</li>
```

```
        <li id="2" routerLink="/magaza/siparis" class="list-item"[routerLinkActiveOptions]="{exact : true}" routerLinkActive="Sitem" >siparişlerim</li>
```

```
        <li id="3" routerLink="/magaza/gecmis" class="list-item"[routerLinkActiveOptions]="{exact : true}" routerLinkActive="Sitem" >Sipariş geçmişi</li>
```

```
    </ul>
```

```
</div>
```

```
<div class="font">
```

```
    <div class="card-bg mb-4 px-2 mx-4 text-white ">
```

```
    <div class="table-responsive max-overflow">
```

```
        <table class="table table-borderless text-center text-white ">
```

```

<thead>

<tr>

<th class="text" scope="col">Ürün id</th>

<th class="text" scope="col">Ürün ad</th>

<th class="text" scope="col">Kategori</th>

<th class="text" scope="col">Marka</th>

<th class="text" scope="col">Adet</th>

<th class="text" scope="col">Firma</th>

<th class="text" scope="col">Tutar</th>

<th class="text" scope="col">Sipariş durum</th>

<th class="text" scope="col">İşlem</th>

</tr>

</thead>

<tbody class="w-100"><!--ngFor-->

<tr class="" *ngFor="let urun of siparisler">

<td class="text col text-break" scope="row">{{ urun.urun_ID }}</td>

<td class="text col-2 text-break">{{ urun.urun_ad }}</td>

<td class="text col text-break">{{ urun.kategori_AD }}</td>

<td class="text col text-break">{{ urun.marka_AD }}</td>

<td class="text col text-break">{{ urun.siparis_adeti }}</td>

<td class="text col text-break">AH firması</td>

<td class="text col text-break">{{ (urun.urun_adet*urun.satis) }} TL</td>

<td *ngIf="urun.durum=='sipariş hazırlanıyor'" class="text col text-break
yesil">{{ urun.durum }}</td>

<td *ngIf="urun.durum=='sipariş bekliyor'" class="text col text-break sari">{{ urun.durum }}</td>

<td class="text col">

<button id={{ urun.id+urun.firma }} (click)="sil(urun)" type="button" class="btn btn-sm text-white
h-100 btn-danger">Kaldır</button>

</td>

```

```
</tr>

</tbody>

</table>

</div>


</div>


</div>
```

---

## 7.Yönlendirmeler ve Etkileşimler (TS kodları)

### ----app.routing.module.ts----

```
import { NgModule } from '@angular/core';
import { RouterModule, Routes } from '@angular/router';
import { ContentComponent } from './content/content.component';
import { MainComponent } from './main/main.component';
import { OptionsComponent } from './main/options/options.component';
import { ProductStockComponent } from './main/product-stock/product-stock.component';
import { ProductComponent } from './main/product/product.component';
import { SellCalendarComponent } from './islemci/sell-calendar/sell-calendar.component';
import { Component, OnInit, ElementRef, Renderer2 } from '@angular/core';
import { FormControl, FormGroup, Validators } from '@angular/forms';
import { AuthService } from 'src/app/services/auth.service';
import { HataComponent } from './hata/hata.component';
import { MagazaComponent } from './main/magaza/magaza.component';
import { SepetComponent } from './main/magaza/sepet/sepet.component';
import { GecmisComponent } from './main/magaza/gecmis/gecmis.component';
import { SiparislerComponent } from './main/magaza/siparisler/siparisler.component';
import { DepoComponent } from './depo/depo.component';
import { IslemciComponent } from './islemci/islemci/islemci.component';
import { EkleComponent } from './islemci/ekle/ekle.component';
```

```

import { GecmisComponent2 } from './islemci/gecmis/gecmis.component';
import { HazirlananComponent } from './islemci/hazirlanan/hazirlanan.component';
import { KayitComponent } from './depo/kayit/kayit.component';
import { YonetRaporComponent } from './yonetici/yonet-rapor/yonet-rapor.component';
import { YoneticiComponent } from './yonetici/yonetici.component';
import { YonetGecmisComponent } from './yonetici/yonet-gecmis/yonet-gecmis.component';
import { YonetHesapComponent } from './yonetici/yonet-hesap/yonet-hesap.component';
export const routes: Routes = [

  {path:"",component:ContentComponent},
  {path:'main',component:YoneticiComponent } ,
  {path:'main/rapor',component:YonetRaporComponent},
  {path:'main/recently', component:YonetGecmisComponent},
  {path:'main/hesap', component:YonetHesapComponent},
  {path:'islemci',component:IslemciComponent},
  {path:'islemci/ekle',component:EkleComponent},
  {path:'islemci/hazirlanan',component:HazirlananComponent},
  {path:'islemci/gecmis',component:GecmisComponent2},
  {path:'depo/stock',component:ProductStockComponent},
  {path:'hata',component:HataComponent},
  {path:'magaza',component:MagazaComponent},
  {path:'magaza/sepet',component:SepetComponent},
  {path:'magaza/gecmis',component:GecmisComponent},
  {path:'magaza/siparis',component:SiparislerComponent},
  {path:'depo',component:DepoComponent},
  {path:'depo/kayit',component:KayitComponent},
  {path:'**',component:HataComponent}
];
@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})

```

```
export class AppRoutingModule { }
```

-----

#### **----app.module.ts----**

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppRoutingModule } from './app-routing.module';
import { AppComponent } from './app.component';
import { NavbarComponent } from './navbar/navbar.component';
import { ContentComponent } from './content/content.component';
import { MainComponent } from './main/main.component';
import { AnalysisGraphicsComponent } from './main/analysis-graphics/analysis-graphics.component';
import { ProductStockComponent } from './main/product-stock/product-stock.component';
import { ProductComponent } from './main/product/product.component';
import { OptionsComponent } from './main/options/options.component';
import { SellCalendarComponent } from './islemci/sell-calendar/sell-calendar.component';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';
import { HataComponent } from './hata/hata.component';
import { routes } from './app-routing.module';
import { ReactiveFormsModule } from '@angular/forms';
import { HttpClientModule } from '@angular/common/http';
import { AuthService } from './services/auth.service';
import { HttpClient } from '@angular/common/http';
import { Injectable } from '@angular/core';
import { RouterModule } from '@angular/router';
import { MagazaComponent } from './main/magaza/magaza.component';
import { SepetComponent } from './main/magaza/sepet/sepet.component';
import { SiparislerComponent } from './main/magaza/siparisler/siparisler.component';
import { GecmisComponent } from './main/magaza/gecmis/gecmis.component';
import { DepoComponent } from './depo/depo.component';
import { IslemciComponent } from './islemci/islemci/islemci.component';
import { EkleComponent } from './islemci/ekle/ekle.component';
```

```
import { GecmisComponent2 } from './islemci/gecmis/gecmis.component';

import { HazirlananComponent } from './islemci/hazirlanan/hazirlanan.component';

import { KayitComponent } from './depo/kayit/kayit.component';

import { FormsModule } from '@angular/forms';

import { YoneticiComponent } from './yonetici/yonetici.component';

import { YonetRaporComponent } from './yonetici/yonet-rapor/yonet-rapor.component';

import { YonetGecmisComponent } from './yonetici/yonet-gecmis/yonet-gecmis.component';

import { YonetHesapComponent } from './yonetici/yonet-hesap/yonet-hesap.component';

@NgModule({
  declarations: [
    AppComponent,
    NavbarComponent,
    ContentComponent,
    MainComponent,
    AnalysisGraphicsComponent,
    ProductStockComponent,
    ProductComponent,
    HataComponent,
    OptionsComponent,
    SellCalendarComponent,
    MagazaComponent,
    SepetComponent,
    SiparislerComponent,
    GecmisComponent,
    DepoComponent,
    IslemciComponent,
    EkleComponent,
    GecmisComponent2,
    HazirlananComponent,
    KayitComponent,
    YoneticiComponent,
```

```

    YonetRaporComponent,
    YonetGecmisComponent,
    YonetHesapComponent,
  ],
  imports: [
    BrowserModule,
    AppRoutingModule,
    BrowserModuleAnimationsModule,
    ReactiveFormsModule,
    HttpClientModule,
    RouterModule.forRoot(routes) // routes değişkenini forRoot() fonksiyonuna parametre olarak gönderiyoruz

  ],
  providers: [
    AuthService
  ],
  bootstrap: [AppComponent]
})
export class AppModule { }

```

#### -----yonetici.component.ts-----

```

import { Component, ElementRef, ViewChild, OnInit, Renderer2 } from '@angular/core';
import { FormControl, Validators } from '@angular/forms';
import { Router } from '@angular/router';

import { Chart , Colors, registerables } from 'node_modules/chart.js';
import { AuthService } from '../services/auth.service';

Chart.register(...registerables);

```

```
@Component({
  selector: 'app-yonetici',
  templateUrl: './yonetici.component.html',
  styleUrls: ['./yonetici.component.css']
})
export class YoneticiComponent {
```

```
  tip:any='bar';
  group !:Element;
  pie_ch !:Element;
  graphics !:Element;
  lyear !:Element;
  lmonth !:Element;
  lweek !:Element;
  graph !:any;
  analiz !:Element;
  chart!:Chart;
  baslik!:Element;
  loginn!:Element;
  blocker!:Element;
  azcok !:Element;
  cokaz !:Element;
  st !:Element;
  ln !:Element;
  today !:Element;
  startDateControl: FormControl;
  endDateControl: FormControl;
  kat!:any;
  mark!:any;
  //tavsiyeler
  tavsiyeler:any[][]=[];
```



```
tavsiyelerCopy:any=[];

//grafik

grafik:any=[];

//sepettekiler

sepettekiler:any=[];

sepettekilerCopy:any=[];

//satanlar

kategoriler:any=[];

markalar:any=[];

satanlar:any=[];

analiztab:any=[];

satanlarCopy:any=[];

ikihafta: any=[];

analiztab1: any=[];

constructor(public router: Router,private elementRef: ElementRef, private renderer: Renderer2,private
service:AuthService) {

    this.startDateControl = new FormControl(null, Validators.required);

    this.endDateControl = new FormControl(null, Validators.required);

}

deger="satan";


ozel!:Element;


ngOnInit(){

    if ((localStorage.getItem('ls')== '1')) {

        // magazalogin true ise

    } else {

        // magazalogin false ise

        this.router.navigate(['/hata']);

    }

}

//sepet gecmiş
```

```
this.service.getsepetgecmis().subscribe(

  (data) => {

    this.sepettekiler=data;


  },

  (error) => {

    console.error(error);

    // Hata işlemlerini burada gerçekleştirin

  }

);

//getsatisgunlerbirlesik

this.service.getsatisgunlerbirlesik().subscribe(

  (data) => {

    this.satanlar=data;


  },

  (error) => {

    console.error(error);

    // Hata işlemlerini burada gerçekleştirin

  }

);

this.service.tavsiye().subscribe(

  (data) => {

    this.tavsiyeler=data;

    const buttonElement = this.elementRef.nativeElement.querySelector('#today');

    buttonElement.click();

    this.zaman="today";

    this.tavsiye();


  },
```

```
(error) => {  
    console.error(error);  
    // Hata işlemlerini burada gerçekleştirin  
}  
);  
  
this.service.getkategori().subscribe(  
    (data) => {  
        this.kategoriler=data;  
        const buttonElement = this.elementRef.nativeElement.querySelector('#today');  
        buttonElement.click();  
        this.zaman="today";  
        this.tavsiye();  
  
    },  
    (error) => {  
        console.error(error);  
        // Hata işlemlerini burada gerçekleştirin  
    }  
);  
  
this.service.getmarka().subscribe(  
    (data) => {  
        this.markalar=data;  
        const buttonElement = this.elementRef.nativeElement.querySelector('#today');  
        buttonElement.click();  
        this.zaman="today";  
        this.tavsiye();  
  
    },  
    (error) => {  
        console.error(error);
```

```
// Hata işlemlerini burada gerçekleştirin
}
);

this.mark=document.querySelector('#selectMark')!;
this.kat=document.querySelector('#selectCat')!;
this.cokaz=document.querySelector('#cokaz')!;
this.azcok=document.querySelector('#azcok')!;
this.lweek=document.querySelector('#LastWeek')!;
this.lmonth=document.querySelector('#LastMonth')!;
this.today=document.querySelector('#today')!;
this.lyear=document.querySelector('#LastYear')!;
this.group=document.querySelector('#group')!;;
this.graphics=document.querySelector('#graph')!;
this.graph=document.querySelector('#chart')!;
this.analiz=document.querySelector('#analiz')!;
this.loginn=document.querySelector('#loginn')!;
this.blocker=document.querySelector('.blocker')!;
this.st=document.querySelector('#st')!;
this.ln=document.querySelector('#ln')!;
this.baslik=document.querySelector('#baslik')!;
this.ozel=document.querySelector('#ozel')!;
```

```
this.RenderChart();
```

```
}
```

```
bastarih:any;
```

```
bittarih:any;
```

```
zaman:any;
```

```
submitForm() {  
  
    if(new Date(this.endDateControl.value)>=new Date(this.startDateControl.value))  
  
    if (new Date(this.endDateControl.value)<=new Date()) {  
  
        this.bastarih=this.startDateControl.value;  
  
        this.bittarih=this.endDateControl.value;  
  
        this.zaman="ozel";  
  
        this.filtre();  
  
        this.ozel.classList.add("aktif");  
  
        this.today.classList.remove("aktif");  
this.lweek.classList.remove("aktif");  
this.lyear.classList.remove("aktif");  
this.lmonth.classList.remove("aktif");  
this.loginn.classList.add('d-none');  
this.blocker.classList.add('d-none');  
this.renderer.setStyle(document.body, 'overflow', 'auto');  
  
    } else {  
  
        alert('Bitiş tarihi bugünün tarihinden büyük olamaz!');  
  
    }else{  
  
        alert('Başlangıç tarihi bitiş tarihinden büyük olamaz!');  
  
    }  
  
}
```

```
tavsiye(){  
  
    console.log(this.zaman);  
  
    this.ikihafta=this.tavsiyeler.slice(0);  
  
    this.ikihafta=this.ikihafta.map((item)=>item.degisken);  
  
  
  
    if(this.zaman=="today"){  
  
        this.tavsiyelerCopy=this.tavsiyeler.slice(0, 9);
```

```
        this.tavsiyelerCopy=this.tavsiyelerCopy.map((item)=>item.degisken);
    }
    if(this.zaman=="week"){
        this.tavsiyelerCopy=this.tavsiyeler.slice(9, 17);
        this.tavsiyelerCopy=this.tavsiyelerCopy.map((item)=>item.degisken);

    }
    if(this.zaman=="month"){
        this.tavsiyelerCopy=this.tavsiyeler.slice(17, 25);
        this.tavsiyelerCopy=this.tavsiyelerCopy.map((item)=>item.degisken);

    }
    if(this.zaman=="year"){
        this.tavsiyelerCopy=this.tavsiyeler.slice(25, 34);
        this.tavsiyelerCopy=this.tavsiyelerCopy.map((item)=>item.degisken);

    }
    console.log(this.tavsiyelerCopy)
}
```

```
grafikveustfiltre(bas:any,son:any=new Date()){
```

```
    if(son instanceof Date) {
        son.setHours(0,0,0,0);
        this.satanlarCopy=this.satanlarCopy.filter((item)=> new Date(item.tarih)>=bas);
        this.sepettekilerCopy=this.sepettekilerCopy.filter((item)=> new Date(item.siparis_tarih)>=bas);
    }else{
```

```

son=new Date(son);

son.setHours(0,0,0,0);

this.satanlarCopy=this.satanlarCopy.filter((item)=> new Date(item.tarih)>=bas && son>=new
Date(item.tarih));

this.sepettekilerCopy=this.sepettekilerCopy.filter((item)=> new Date(item.siparis_tarih)>=bas &&
son>=new Date(item.siparis_tarih));

}

```

```

console.log(this.satanlarCopy);

let dizi = [];

let dizi1 = [];

let idler:any[]=[];

if(this.deger=="satan" || this.deger=="kar"){

for (let i = 0; i < this.satanlarCopy.length; i++) {

    let id = this.satanlarCopy[i].urun_ID;

    let ad = this.satanlarCopy[i].urun_ad;

    let tarih = this.satanlarCopy[i].tarih;

    let mal = this.satanlarCopy[i].maliyet;

    let sat = this.satanlarCopy[i].satis;

    let spt = 0;

    for (let j = 0; j < this.satanlarCopy.length; j++) {

        if (id === this.satanlarCopy[j].urun_ID) {

            spt += this.satanlarCopy[j].toplam_adet;

        }

    }

    if(idler.filter((item)=>item==id).length<=0){

        dizi.push({ id: id, ad:ad, tarih: tarih, adet: spt ,mal:mal ,satis:sat, kar:spt*((parseInt(sat))-
(parseInt(mal))))});

        dizi1.push({ id: id, ad:ad, tarih: tarih, adet: spt ,mal:mal ,satis:sat, kar:spt*((parseInt(sat))-
(parseInt(mal))))});

    }

    idler.push(id);

}
}

```

```

if(this.deger=="satan"){

    dizi.sort((a, b) => b.adet - a.adet);

    this.analiztab=dizi.splice(0,dizi.length);

    this.analiztab1=this.analiztab;

    console.log(this.analiztab);

    if(dizi.length>=10)

        this.grafik=dizi1.splice(0,10);

        else

            this.grafik=dizi1;

    }

    if(this.deger=="kar"){

        dizi.sort((a, b) => b.kar - a.kar);

        this.analiztab=dizi.splice(0,dizi.length);

        this.analiztab1=this.analiztab;


        console.log(this.analiztab);

        if(dizi.length>=10)

            this.grafik=dizi1.splice(0,10);

            else

                this.grafik=dizi1;

        }

    }

    else{

        for (let i = 0; i < this.sepettekilerCopy.length; i++) {

            let id = this.sepettekilerCopy[i].urun_ID;

            let ad = this.sepettekilerCopy[i].urun_ad;

            let tarih = this.sepettekilerCopy[i].siparis_tarih;

            let spt = 0;

            for (let j = 0; j < this.sepettekilerCopy.length; j++) {

                if (id === this.sepettekilerCopy[j].urun_ID) {

                    spt += this.sepettekilerCopy[j].adet;

                }

            }

        }

    }

}

```



```
    }  
  }  
  if(idler.filter((item)=>item==id).length<=0){  
    dizi.push({ id: id, ad:ad, tarih: tarih, adet: spt });  
    dizi1.push({ id: id, ad:ad, tarih: tarih, adet: spt });  
  }  
  idler.push(id);  
}
```

```
dizi.sort((a, b) => b.adet - a.adet);  
this.analiztab=dizi.splice(0,dizi.length);  
this.analiztab1=this.analiztab;
```

```
console.log(this.analiztab);  
if(dizi.length>=10)  
  this.grafik=dizi1.splice(0,10);  
else  
  this.grafik=dizi1;
```

```
}
```

```
this.chart.destroy();  
this.RenderChart();  
}
```

```
filtre() {  
  this.chart.destroy();  
  this.RenderChart();  
}
```

```

    const kat = this.kat.value;

    const marka = this.mark.value;

console.log(kat,marka);

if(this.deger=="satan" || this.deger=="kar"){

    if (kat === '0' && marka === '0') {

        this.satanlarCopy = this.satanlar.filter((item)=>item);

    } else if (kat === "0" && marka !== "0") {

        // Sadece marka filtresi uygula

        this.satanlarCopy = this.satanlar.filter(item => parseInt(item.marka_ID) ===parseInt(marka));

    } else if (kat !== "0" && marka === "0") {

        // Sadece kategori filtresi uygula

        this.satanlarCopy = this.satanlar.filter(item => parseInt(item.kategori_ID) ===parseInt(kat));

    } else {

        // Hem kategori hem de marka filtresi uygula

        this.satanlarCopy = this.satanlar.filter(item => item.kategori_ID== parseInt(kat) && item.marka_ID
===parseInt(marka));

    }

}

else{

    if (kat === '0' && marka === '0') {

        this.sepettekilerCopy = this.sepettekiler.filter((item)=>item);

    } else if (kat === "0" && marka !== "0") {

        // Sadece marka filtresi uygula

        this.sepettekilerCopy = this.sepettekiler.filter(item => parseInt(item.marka_ID) ===parseInt(marka));

    } else if (kat !== "0" && marka === "0") {

        // Sadece kategori filtresi uygula

        this.sepettekilerCopy = this.sepettekiler.filter(item => parseInt(item.kategori_ID) ===parseInt(kat));

    } else {

        // Hem kategori hem de marka filtresi uygula

        this.sepettekilerCopy = this.sepettekiler.filter(item => item.kategori_ID== parseInt(kat) && item.marka_ID
===parseInt(marka));

    }

}

```

```
}
```

```
if(this.zaman=="today"){  
    const tarih=new Date();  
    tarih.setHours(0, 0, 0, 0);  
    this.grafikveustfiltre(tarih);  
}
```

```
if(this.zaman=="week"){  
    const tarih=new Date();  
    tarih.setHours(0, 0, 0, 0);  
    tarih.setDate(tarih.getDate()-7)  
    this.grafikveustfiltre(tarih);  
}
```

```
if(this.zaman=="month"){  
    const tarih=new Date();  
    tarih.setHours(0, 0, 0, 0);  
    tarih.setMonth(tarih.getMonth()-1)  
    this.grafikveustfiltre(tarih);  
}
```

```
if(this.zaman=="year"){  
    const tarih=new Date();  
    tarih.setHours(0, 0, 0, 0);  
    tarih.setFullYear(tarih.getFullYear()-1)  
    this.grafikveustfiltre(tarih);  
}
```

```
if(this.zaman=="ozel"){  
    const tarih=new Date(this.bastarih);  
    const tarih1=this.bittarih;  
    tarih.setHours(0, 0, 0, 0);  
    this.grafikveustfiltre(tarih,tarih1);  
}
```

```
}  
}
```

```
ara(event:any){
```

```
    console.log(this.analiztab1);
```

```
    if(this.deger=="satan"||this.deger=="kar"){
```

```
        this.analiztab1=this.analiztab.filter(item =>  
item.ad.toLowerCase().includes(event.target.value.toLowerCase()));
```

```
        console.log(this.analiztab);
```

```
    }
```

```
else
```

```
this.analiztab = this.analiztab1.filter(item => item.ad.includes(event.target.value));
```

```
}
```

```
acc(event:Event){
```

```
    event.preventDefault();
```

```
    this.loginn.classList.remove('d-none');
```

```
    this.blocker.classList.remove('d-none');
```

```
    this.renderer.setStyle(document.body, 'overflow', 'hidden');
```

```
}
```

```
val(event :any){
```

```
this.deger=event.target.value;
```

```
this.baslik.textContent==event.target.textContent;
```

```
this.filtre();
```

```
this.tavsiye();
```

```
event.preventDefault();
```

```
this.chart.destroy();
```

```
this.RenderChart();
```

```
}
```

```
kapatt(event:Event){
```

```
    event.preventDefault();
```

```
    this.loginn.classList.add('d-none');
```

```
    this.blocker.classList.add('d-none');
```

```
    this.renderer.setStyle(document.body, 'overflow', 'auto');
```

```
}
```

```
btnClick(event:Event){
```

```
    console.log(event);
```

```
    event.preventDefault();
```

```
    this.chart.destroy();
```

```
    this.RenderChart();
```

```
if(this.st.classList.contains("aktif")){
```

```
    this.ln.classList.add("aktif");
```

```
    this.st.classList.remove("aktif");
```

```
}
```

```
if(this.ln.classList.contains("aktif")){
```

```
this.st.classList.add("aktif");
```

```
    this.ln.classList.remove("aktif");
```

```
}
```

```
}
```

```
cokAz(){  
  
    this.cokaz.classList.add("aktif");  
  
    this.azcok.classList.remove("aktif");  
  
    if(this.deger=="satan" || this.deger=="sepet")  
  
        this.analiztab.sort((a, b) => b.adet - a.adet);  
  
    else  
  
        this.analiztab.sort((a, b) => b.kar - a.kar);  
  
}
```

```
azCok(){  
  
    this.cokaz.classList.remove("aktif");  
  
    this.azcok.classList.add("aktif");  
  
    if(this.deger=="satan" || this.deger=="sepet")  
  
        this.analiztab.sort((a, b) => a.adet - b.adet);  
  
    else  
  
        this.analiztab.sort((a, b) => a.kar - b.kar);  
  
}
```

```
todayy(event:any){  
  
  
  
    this.ozel.classList.remove("aktif");  
  
    this.lweek.classList.remove("aktif");  
  
    this.lyear.classList.remove("aktif");  
  
    this.lmonth.classList.remove("aktif");  
  
    this.today.classList.add("aktif");  
  
    this.zaman="today";  
  
  
  
    this.filtre();
```

```
this.tavsiye();  
  
event.preventDefault();  
  
this.chart.destroy();  
  
this.RenderChart();  
  
}
```

```
lastWeek(event:any){  
  
    this.ozel.classList.remove("aktif");  
  
this.lweek.classList.add("aktif");  
  
this.lyear.classList.remove("aktif");  
  
this.lmonth.classList.remove("aktif");  
  
this.today.classList.remove("aktif");  
  
this.zaman="week";
```

```
  
  
this.filtre();  
  
event.preventDefault();  
  
    this.chart.destroy();  
  
    this.RenderChart();  
  
this.tavsiye();
```

```
}
```

```
lastMonth(event:any){  
  
    this.ozel.classList.remove("aktif");  
  
    this.today.classList.remove("aktif");  
  
this.lweek.classList.remove("aktif");  
  
this.lyear.classList.remove("aktif");  
  
this.lmonth.classList.add("aktif");  
  
this.zaman="month";
```

```
this.filtre();  
  
this.tavsiye();  
  
event.preventDefault();  
  
    this.chart.destroy();  
  
    this.RenderChart();  
  
}
```

```
lastYear(event:any){  
  
    this.ozel.classList.remove("aktif");  
  
    this.today.classList.remove("aktif");  
  
this.lweek.classList.remove("aktif");  
  
this.lyear.classList.add("aktif");  
  
this.lmonth.classList.remove("aktif");  
  
this.zaman="year";
```

```
this.filtre();  
  
this.tavsiye();  
  
event.preventDefault();  
  
    this.chart.destroy();  
  
    this.RenderChart();  
  
}
```

```
RenderChart(){
```

```
    Chart.defaults.backgroundColor = '#a233';  
  
    Chart.defaults.borderColor = 'rgba(239, 238, 238, 0.5)';  
  
    Chart.defaults.color = 'white';  
  
    Chart.defaults.font.size=15;  
  
    Chart.defaults.responsive=true;
```



```

this.chart=new Chart(this.graph, {
    type:this.tip,
    data: {
        labels:this.grafik.map(item => item.ad)
    },
    datasets: [{
        label: 'Ürünler',
        data:this.grafik.map(item => item.adet),
        borderWidth: 1,
        hoverRadius:12,
        pointRadius:7,
        backgroundColor:['rgba(198, 198, 198, 0.3)'],
    }]
},
options: {
    scale: {
        yAxes: [{
            ticks: {

            }
        }]
    }
});

```

#### ----yonet-rapor.component.ts----

```

import { Component, ElementRef, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { jsPDF } from 'jspdf';
import { AuthService } from 'src/app/services/auth.service';

```

```
import { addEmitHelper } from 'typescript';

@Component({
  selector: 'app-yonet-rapor',
  templateUrl: './yonet-rapor.component.html',
  styleUrls: ['./yonet-rapor.component.css']
})
```

```
export class YonetRaporComponent implements OnInit{
```

```
  enazsatan: string | undefined;
```

```
  enazsatankez: number | undefined;
```

```
  enazkat: string | undefined;
```

```
  enazkatkez: number | undefined;
```

```
  enazmark: string | undefined;
```

```
  enazmarkkez: number | undefined;
```

```
  constructor(private service: AuthService,private router:Router ,private elementRef:ElementRef) { }
```

```
  lyear !:Element;
```

```
  lmonth !:Element;
```

```
  lweek !:Element;
```

```
  siparisler:any[]=[];
```

```
  gecmis :any[]=[];
```

```
  alinanlar:any[]=[];
```

```
  zaman="week";
```

```
  totalal:any;
```

```
  totalsat:any;
```

```
  topmal:any;
```

```
  topkaz:any;
```

```
  totalsip:any;
```

```
  day:any;
```

```
  month:any;
```

```
  year:any;
```

basarili:any;  
basarisiz:any;  
satan:any;  
satankez:any;  
enmark:any;  
markkez:any;  
enkat:any;  
katkez:any;  
teslimler:any;  
month1!:any;  
day1!:any;  
lday!:any;

ngOnInit() {

```
if ((localStorage.getItem('ls')=='1')) {  
    // magazalogin true ise  
} else {  
    // magazalogin false ise  
    this.router.navigate(['/hata']);  
}
```

```
this.lday=document.querySelector('#LastDay')!;  
this.lweek=document.querySelector('#LastWeek')!;  
this.lmonth=document.querySelector('#LastMonth')!;  
this.lyear=document.querySelector('#LastYear')!;  
this.day=new Date().getDate();  
this.day1=new Date().getDate();  
this.month=new Date().getMonth();  
this.month1=new Date().getMonth();
```

```
this.year=new Date().getFullYear();
```

```
this.day1=this.day1-7;
```

```
this.month=this.month+1;
```

```
if(this.day1<10){
```

```
    this.day1="0"+this.day1.toString();
```

```
    this.day1=parseInt(this.day1);
```

```
}
```

```
const buttonElement = this.elementRef.nativeElement.querySelector('#LastWeek');
```

```
buttonElement.click();
```

```
if(this.month1+1<10){
```

```
    this.month1 = (this.month1).toString().padStart(2, "0");
```

```
}
```

```
else{
```

```
    this.month=this.month;
```

```
}
```

```
}
```

```
PDFDownload(){
```

```
    const doc = new jsPDF();
```

```
    doc.addFont('assets/fonts/BebasNeue-Regular.ttf', 'ArialUnicodeMS', 'normal');
```

```
    doc.setFont('ArialUnicodeMS');
```

```
const element = document.getElementById('pdfContent')!;

const htmlContent = element.innerHTML;


const plainText = htmlContent.replace(/<[^>]+>/g, "");


const lines = plainText.split('\n');

const lineHeight = 10;


let y = 10;


lines.forEach(line => {

  doc.text(line, 10, y);

  y += lineHeight;

});


doc.save('dosya.pdf');

}


async a(event:any){

  try {

    event.preventDefault();


    const data1 = await this.service.getalis().toPromise();

    this.alinanlar = data1!;


    const data2 = await this.service.getsiparislerim().toPromise();

    this.siparisler = data2!;
```

```
const data3 = await this.service.gecmis().toPromise();

this.gecmis = data3!;

console.log(this.gecmis);

this.lastWeek();

} catch (error) {

    console.error(error);

    // Hata işlemlerini burada gerçekleştirin

}

};

b(){};

c(){};

lastWeek(){

    if(this.month+1<10){

        this. month="0"+(this.month);

    }

    else{

        this.month=this.month;

    }

}

this.topkaz=0;

this.topmal=0;

this.totalal=0;

this.totalsat=0;

const ortsure=0;

this.zaman="week";

var sepet=0;

const hafta1=this.gecmis.filter((item)=>item.siparis_tarih>=`${this.year}-${this.month}-${this.day1}T00:00:00`);

for (let i = 0; i < hafta1.length; i++) {
```

```
const siparis = hafta1[i].siparis_tarih;

const teslim = hafta1[i].teslim_tarihi;


const tarih1 = new Date(siparis);

const tarih2 = new Date(teslim);


const tarih1ZamanDamgasi = tarih1.getTime();

const tarih2ZamanDamgasi = tarih2.getTime();


var gunFarki = Math.floor((tarih2ZamanDamgasi - tarih1ZamanDamgasi) / (1000 * 60 * 60 * 24));

sepet+=gunFarki;

}

console.log(sepet/this.gecmis.length);


this.teslimler = (sepet / this.gecmis.length).toFixed(1);


this.gecmis

console.log(this.year,this.month,this.day1);

const hafta=this.gecmis.filter((item)=>item.siparis_tarih>=`${this.year}-${this.month}-${this.day1}T00:00:00` && item.durum_id!=0);

console.log(hafta);

for(var i=0;i<hafta.length;i++){

this.topkaz+=hafta[i].satis*hafta[i].adet;

this.topmal+=hafta[i].adet*hafta[i].maliyet;

this.totalsat+=hafta[i].adet;

}

const alinanlar=this.alinanlar.filter((item)=>item.tarih>=`${this.year}-${this.month}-${this.day1}T00:00:00`)

for(var i=0;i<alinanlar.length;i++){
```

```

        this.totalal+=alınanlar[i].adet;

    }

    const basarili=this.gecmis.filter((item)=>item.siparis_tarih>=`${this.year}-${this.month}-${this.day1}T00:00:00` && item.durum_id==3);

    const basarisiz=this.gecmis.filter((item)=>item.siparis_tarih>=`${this.year}-${this.month}-${this.day1}T00:00:00` && item.durum_id==0);

    this.basarili=basarili.length;

    this.basarisiz=basarisiz.length;

    this.totalsip=hafta.length+basarisiz.length;

    this.findMostSoldProduct(hafta);

    this.findLeastSoldProduct(hafta);

    console.log(this.basarili,this.basarisiz);

    this.lweek.classList.add("aktif");

    this.lyear.classList.remove("aktif");

    this.lmonth.classList.remove("aktif");

    this.lday.classList.remove("aktif");

}

```

```

findMostSoldProduct(salesRecords:any[]) {

    var productSalesCounter:any = { };

    var categorySalesCounter:any = { };

    var brandSalesCounter:any = { };

    for (var i = 0; i < salesRecords.length; i++) {

        var product = salesRecords[i].urun_ad;

        var category = salesRecords[i].kategori_AD;

        var brand = salesRecords[i].marka_AD;

        var quantity = salesRecords[i].adet;
    }
}

```



```
// Ürün satışlarını takip et

if (productSalesCounter[product]) {

    productSalesCounter[product] += quantity;

} else {

    productSalesCounter[product] = quantity;

}


// Kategori satışlarını takip et

if (categorySalesCounter[category]) {

    categorySalesCounter[category] += quantity;

} else {

    categorySalesCounter[category] = quantity;

}


// Marka satışlarını takip et

if (brandSalesCounter[brand]) {

    brandSalesCounter[brand] += quantity;

} else {

    brandSalesCounter[brand] = quantity;

}

}

var mostSoldProduct = "";

var mostSoldProductSales = 0;


for (let product in productSalesCounter) {

    if (productSalesCounter[product] > mostSoldProductSales) {

        mostSoldProductSales = productSalesCounter[product];

        mostSoldProduct = product;

    }

}
```

```
var mostSoldCategory = "";
var mostSoldCategorySales = 0;

for (let category in categorySalesCounter) {
  if (categorySalesCounter[category] > mostSoldCategorySales) {
    mostSoldCategorySales = categorySalesCounter[category];
    mostSoldCategory = category;
  }
}
```

```
var mostSoldBrand = "";
var mostSoldBrandSales = 0;
```

```
for (let brand in brandSalesCounter) {
  if (brandSalesCounter[brand] > mostSoldBrandSales) {
    mostSoldBrandSales = brandSalesCounter[brand];
    mostSoldBrand = brand;
  }
}
```

```
this.satan=mostSoldProduct;
this.satankez=mostSoldProductSales;
this.enkat=mostSoldCategory;
this.katkez=mostSoldCategorySales;
this.enmark= mostSoldBrand;
this.markkez=mostSoldBrandSales;
}

findLeastSoldProduct(salesRecords: any[]) {
```

```
var productSalesCounter: any = { };
var categorySalesCounter: any = { };
var brandSalesCounter: any = { };

for (var i = 0; i < salesRecords.length; i++) {
    var product = salesRecords[i].urun_ad;
    var category = salesRecords[i].kategori_AD;
    var brand = salesRecords[i].marka_AD;
    var quantity = salesRecords[i].adet;

    // Ürün satışlarını takip et
    if (productSalesCounter[product]) {
        productSalesCounter[product] += quantity;
    } else {
        productSalesCounter[product] = quantity;
    }

    // Kategori satışlarını takip et
    if (categorySalesCounter[category]) {
        categorySalesCounter[category] += quantity;
    } else {
        categorySalesCounter[category] = quantity;
    }

    // Marka satışlarını takip et
    if (brandSalesCounter[brand]) {
        brandSalesCounter[brand] += quantity;
    } else {
        brandSalesCounter[brand] = quantity;
    }
}
```

```
var leastSoldProduct = "";
var leastSoldProductSales = Infinity;

for (let product in productSalesCounter) {
  if (productSalesCounter[product] < leastSoldProductSales && productSalesCounter[product] > 0) {
    leastSoldProductSales = productSalesCounter[product];
    leastSoldProduct = product;
  }
}

var leastSoldCategory = "";
var leastSoldCategorySales = Infinity;

for (let category in categorySalesCounter) {
  if (categorySalesCounter[category] < leastSoldCategorySales && categorySalesCounter[category] > 0) {
    leastSoldCategorySales = categorySalesCounter[category];
    leastSoldCategory = category;
  }
}

var leastSoldBrand = "";
var leastSoldBrandSales = Infinity;

for (let brand in brandSalesCounter) {
  if (brandSalesCounter[brand] < leastSoldBrandSales && brandSalesCounter[brand] > 0) {
    leastSoldBrandSales = brandSalesCounter[brand];
    leastSoldBrand = brand;
  }
}
```

```
this.enazsatan = leastSoldProduct;  
  
this.enazsatankez = leastSoldProductSales;  
  
this.enazkat = leastSoldCategory;  
  
this.enazkatkez = leastSoldCategorySales;  
  
this.enazmark = leastSoldBrand;  
  
this.enazmarkkez = leastSoldBrandSales;  
  
}
```

```
lastMonth(event:any){  
    event.preventDefault();
```

```
if(this.day<10){  
    this.day="0"+this.day.toString();  
}
```

```
this.topkaz=0;  
  
this.topmal=0;  
  
this.totalal=0;  
  
this.totalsat=0;  
  
const ortsure=0;  
  
  
this.zaman="month";  
  
var sepet=0;  
  
const hafta1=this.gecmis.filter((item)=>item.siparis_tarih>=`${this.year}-${this.month1}-${this.day}T00:00:00`);  
  
for (let i = 0; i < hafta1.length; i++) {  
    const siparis = hafta1[i].siparis_tarih;
```

```

const teslim = hafta1[i].teslim_tarihi;

const tarih1 = new Date(siparis);
const tarih2 = new Date(teslim);

const tarih1ZamanDamgasi = tarih1.getTime();
const tarih2ZamanDamgasi = tarih2.getTime();

var gunFarki = Math.floor((tarih2ZamanDamgasi - tarih1ZamanDamgasi) / (1000 * 60 * 60 * 24));
sepet+=gunFarki;
}
console.log(sepet/this.gecmis.length);

this.teslimler = (sepet / this.gecmis.length).toFixed(1);

this.gecmis
console.log(this.year,this.month1,this.day);

const hafta=this.gecmis.filter((item)=>item.siparis_tarih>=`${this.year}-${this.month1}-${this.day}T00:00:00` && item.durum_id!=0);

console.log(hafta);

for(var i=0;i<hafta.length;i++){

this.topkaz+=hafta[i].satis*hafta[i].adet;

this.topmal+=hafta[i].adet*hafta[i].maliyet;

this.totalsat+=hafta[i].adet;

}

const alinanlar=this.alinanlar.filter((item)=>item.tarih>=`${this.year}-${this.month1}-${this.day}T00:00:00`)

for(var i=0;i<alinanlar.length;i++){

this.totalal+=alinanlar[i].adet;

}

```

```
const basarili=this.gecmis.filter((item)=>item.siparis_tarih>=`${this.year}-${this.month1}-${this.day}T00:00:00` && item.durum_id==3);
```

```
const basarisiz=this.gecmis.filter((item)=>item.siparis_tarih>=`${this.year}-${this.month1}-${this.day}T00:00:00` && item.durum_id==0);
```

```
this.basarili=basarili.length;
```

```
this.basarisiz=basarisiz.length;
```

```
this.totalsip=hafta.length+basarisiz.length;
```

```
this.findMostSoldProduct(hafta);
```

```
this.findLeastSoldProduct(hafta);
```

```
console.log(this.basarili,this.basarisiz);
```

```
this.lweek.classList.remove("aktif");
```

```
this.lyear.classList.remove("aktif");
```

```
this.lmonth.classList.add("aktif");
```

```
this.lday.classList.remove("aktif");
```

```
}
```

```
lastYear(event:any){
```

```
    event.preventDefault();
```

```
    if(this.day<10){
```

```
        this.day="0"+this.day.toString();
```

```
    }
```

```
    if(this.month+1<10){
```

```
        this.month="0"+(this.month);
```

```
    }
```

```
    else{
```

```
        this.month=this.month;
```

```
    }
```

```
console.log(this.day);
```

```
    this.topkaz=0;
```

```

this.topmal=0;

this.totalal=0;

this.totalsat=0;

const ortsure=0;


this.zaman="year";

var sepet=0;

const hafta1=this.gecmis.filter((item)=>item.siparis_tarih>=`${this.year-1}-${this.month}-${this.day}T00:00:00`);

for (let i = 0; i < hafta1.length; i++) {

    const siparis = hafta1[i].siparis_tarih;

    const teslim = hafta1[i].teslim_tarihi;


    const tarih1 = new Date(siparis);

    const tarih2 = new Date(teslim);


    const tarih1ZamanDamgasi = tarih1.getTime();

    const tarih2ZamanDamgasi = tarih2.getTime();


    var gunFarki = Math.floor((tarih2ZamanDamgasi - tarih1ZamanDamgasi) / (1000 * 60 * 60 * 24));

    sepet+=gunFarki;

}

console.log(sepet/this.gecmis.length);


this.teslimler = (sepet / this.gecmis.length).toFixed(1);


this.gecmis

console.log(this.year,this.month,this.day);

const hafta=this.gecmis.filter((item)=>item.siparis_tarih>=`${this.year-1}-${this.month}-${this.day}T00:00:00` && item.durum_id!=0);

console.log(hafta);

```



```

for(var i=0;i<hafta.length;i++){

    this.topkaz+=hafta[i].satis*hafta[i].adet;

    this.topmal+=hafta[i].adet*hafta[i].maliyet;

    this.totalsat+=hafta[i].adet;

}

const alinanlar=this.alinanlar.filter((item)=>item.tarih>=`${this.year-1}-${this.month}-${this.day}T00:00:00`)

for(var i=0;i<alinanlar.length;i++){

    this.totalal+=alinanlar[i].adet;

}

const basarili=this.gecmis.filter((item)=>item.siparis_tarih>=`${this.year-1}-${this.month}-${this.day}T00:00:00` && item.durum_id==3);

const basarisiz=this.gecmis.filter((item)=>item.siparis_tarih>=`${this.year-1}-${this.month}-${this.day}T00:00:00` && item.durum_id==0);

this.basarili=basarili.length;

this.basarisiz=basarisiz.length;

this.totalsip=hafta.length+basarisiz.length;

this.findMostSoldProduct(hafta);

this.findLeastSoldProduct(hafta);

console.log(this.basarili,this.basarisiz);

this.lweek.classList.remove("aktif");

this.lyear.classList.add("aktif");

this.lday.classList.remove("aktif");


this.lmonth.classList.remove("aktif");

}

lastDay(event:any){

    event.preventDefault();

    if(this.day<10){

        this.day="0"+this.day.toString();
    }
}

```

```

    }

    if(this.month+1<10){
        this. month="0"+(this.month);
    }
    else{
        this.month=this.month;
    }
    console.log(this.day);

    this.topkaz=0;
    this.topmal=0;
    this.totalal=0;
    this.totalsat=0;
    const ortsure=0;

    this.zaman="day";
    var sepet=0;

    const hafta1=this.gecmis.filter((item)=>item.siparis_tarih>=`${this.year}-${this.month}-${this.day-1}T00:00:00`);

    for (let i = 0; i < hafta1.length; i++) {
        const siparis = hafta1[i].siparis_tarih;
        const teslim = hafta1[i].teslim_tarihi;

        const tarih1 = new Date(siparis);
        const tarih2 = new Date(teslim);

        const tarih1ZamanDamgasi = tarih1.getTime();
        const tarih2ZamanDamgasi = tarih2.getTime();

        var gunFarki = Math.floor((tarih2ZamanDamgasi - tarih1ZamanDamgasi) / (1000 * 60 * 60 * 24));
        sepet+=gunFarki;
    }

```

```
console.log(sepet/this.gecmis.length);

this.teslimler = (sepet / this.gecmis.length).toFixed(1);

this.gecmis

console.log(this.year,this.month,this.day);

const hafta=this.gecmis.filter((item)=>item.siparis_tarih>=`${this.year}-${this.month}-${this.day-1}T00:00:00` && item.durum_id!=0);

console.log(hafta);

for(var i=0;i<hafta.length;i++){

this.topkaz+=hafta[i].satis*hafta[i].adet;

this.topmal+=hafta[i].adet*hafta[i].maliyet;

this.totalsat+=hafta[i].adet;

}

const alinanlar=this.alinanlar.filter((item)=>item.tarih>=`${this.year}-${this.month}-${this.day-1}T00:00:00`)

for(var i=0;i<alinanlar.length;i++){

this.totalal+=alinanlar[i].adet;

}

const basarili=this.gecmis.filter((item)=>item.siparis_tarih>=`${this.year}-${this.month}-${this.day-1}T00:00:00` && item.durum_id==3);

const basarisiz=this.gecmis.filter((item)=>item.siparis_tarih>=`${this.year}-${this.month}-${this.day-1}T00:00:00` && item.durum_id==0);

this.basarili=basarili.length;

this.basarisiz=basarisiz.length;

this.totalsip=hafta.length+basarisiz.length;

this.findMostSoldProduct(hafta);

this.findLeastSoldProduct(hafta);

console.log(this.basarili,this.basarisiz);

this.lweek.classList.remove("aktif");
```

```
        this.lyear.classList.remove("aktif");

        this.lday.classList.add("aktif");

        this.lmonth.classList.remove("aktif");

    }

}
```

#### ----hazirlanan.component.ts----

```
import { Component, OnInit } from '@angular/core';

import { Router } from '@angular/router';

import { AuthService } from 'src/app/services/auth.service';

@Component({
  selector: 'app-hazirlanan',
  templateUrl: './hazirlanan.component.html',
  styleUrls: ['./hazirlanan.component.css']
})

export class HazirlananComponent implements OnInit {

  constructor(public router : Router, public service : AuthService ) {} ;

  siparisler:any=[];

  sip:any=[];

  veriler:any=[];

  ustBilesenVeri:any=[];

  ngOnInit(): void {

    if (this.service.getMagazallogin()&&(localStorage.getItem('ls')=='2')) {

      // magazallogin true ise

    } else {

      // magazallogin false ise
```

```
this.router.navigate(['/hata']);  
  
}  
  
this.service.hazirlanan().subscribe(  
  
  (data) => {  
  
    this.siparisler = data;  
  
    this.veriler = data;  
  
  
  
    // İstedığınız işlemleri burada gerçekleştirin  
  
  },  
  
  (error) => {  
  
    console.error(error);  
  
    // Hata işlemlerini burada gerçekleştirin  
  
  }  
  
);  
  
this.sip=this.siparisler;  
  
this.veriler=this.sip;  
  
this.ustBilesenVeri = this.sip ;  
  
}  
  
sil(urun:any){  
  
  const siparisadet =parseInt(urun.siparis_adeti); // Siparişin ürün ID'si  
  
  
  
  
  
  
  const siparisid =parseInt(urun.siparis_id); // Siparişin ürün ID'si  
  
  //const adet = parseInt( this.adet.value); // Sipariş adeti  
  
  this.service.sepetsil(siparisid,siparisadet).subscribe(  
  
    () => {  
  
      console.log('siparis silindi.');  
  
      location.reload();  
  
    },  
  
    (error) => {  
  
      console.error('siparis silme hatası:', error);  
  
    }  
  
  }  
  
}
```

```

    }
  );
}

handleSelectedDate(selectedDate: string) {
  console.log(selectedDate);
  console.log(this.siparisler);
  this.veriler=this.siparisler.filter((item)=>item.siparis_tarih==selectedDate);
}

```

```

  Date(veri:any[]){
    this.veriler=veri;
  }
}

```

-----ekle.component.ts-----

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { AuthService } from 'src/app/services/auth.service';

```

```

@Component({
  selector: 'app-ekle',
  templateUrl: './ekle.component.html',
  styleUrls: ['./ekle.component.css']
})
export class EkleComponent implements OnInit {
  constructor(public router : Router, public service : AuthService ) {}

  ngOnInit(): void {
    if (this.service.getMagazalagin()&&(localStorage.getItem('ls')=='2')) {
      // magazalagin true ise
    } else {

```

```
// magazalogin false ise  
  
this.router.navigate(['/hata']);  
  
} } }
```

---

#### ----gecmis.component.ts----

```
import { Component, OnInit } from '@angular/core';  
import { Router } from '@angular/router';  
import { AuthService } from 'src/app/services/auth.service';
```

```
@Component({  
  selector: 'app-gecmis-islemci',  
  templateUrl: './gecmis.component.html',  
  styleUrls: ['./gecmis.component.css']  
})  
  
export class GecmisComponent2 implements OnInit {  
  constructor(public router : Router, public service : AuthService ) {};  
  
  ngOnInit(): void {  
    if (this.service.getMagazalogin()&&(localStorage.getItem('ls')==2')) {  
      // magazalogin true ise  
    } else {  
      // magazalogin false ise  
      this.router.navigate(['/hata']);  
    }  
  }  
}
```

---

#### ----sell-calendar.component.ts----

```
import { Component, EventEmitter, Input, OnInit, Output } from '@angular/core';  
import { Router } from '@angular/router';  
import { AuthService } from 'src/app/services/auth.service';
```

```

@Component({
  selector: 'app-sell-calendar',
  templateUrl: './sell-calendar.component.html',
  styleUrls: ['./../custom_sources/main.css']
})

export class SellCalendarComponent implements OnInit {

  @Output() selectedDateEvent: EventEmitter<string> = new EventEmitter<string>();
  @Output() DateEvent: EventEmitter<Array<any[]>> = new EventEmitter<Array<any[]>>();

  ileri !:Element
  geri !:any
  tbod !:Element
  startDate !: any
  ay !:any;
  tak !:Element;
  calendar: number[][];
  bugun!:Date;

  constructor(private service: AuthService) {
    this.calendar = [];
    this.startDate = new Date();
  }

  veri: any[][]=[];

  ngOnInit(): void {
    this.service.hazirlanan().subscribe(
      (data) => {

```



```

    this.veri=data;

    },

    (error) => {

        console.error(error);

        // Hata işlemlerini burada gerçekleştirin

    }

);

console.log(this.veri);

this.startDate = new Date();

this.startDate.setMonth(this.startDate.getMonth()-1);

this.bugun=new Date();


this.tbod=document.querySelector('#tbod')!;
this.ileri=document.querySelector('#ileri')!;
this.geri=document.querySelector('#geri')!;
this.tak=document.querySelector('#callendar')!;


this.next();

}


i=0;
c:any=0;


isSmallSipDay(sut :any) {
    if(sut<10){
        sut="0"+sut.toString();
    }

    let month:any;

    if(this.bugun.getMonth()+1<10){
        month="0"+(this.bugun.getMonth()+1);
    }
}

```

```
}  
  
else{  
  
    month=this.bugun.getMonth()+1;  
  
}
```

```
const filteredItems = this.veri.filter((item) => item.siparis_tarih == `${this.bugun.getFullYear()}-${month}-${sut}T00:00:00` && this.startDate.getMonth()===this.bugun.getMonth() && this.bugun.getDate()>sut);
```

```
if (filteredItems.length > 0) {
```

```
    return true;  
}
```

```
    return false;  
}
```

```
isSipDay(sut:any) {
```

```
    if(sut<10){  
  
        sut="0"+sut.toString();  
  
    }
```

```
    let month:any;
```

```
    if(this.bugun.getMonth()+1<10){  
  
        month="0"+(this.bugun.getMonth()+1);  
  
    }
```

```
    else{  
  
        month=this.bugun.getMonth()+1;  
  
    }
```

```
const filteredItems = this.veri.filter((item) => item.siparis_tarih == `${this.bugun.getFullYear()}-${this.bugun.getMonth()}-${this.bugun.getDate()}` && this.startDate.getMonth()===this.bugun.getMonth() && new Date().getDate()<=sut);
```

```
if (filteredItems.length > 0) {
```

```
    return true;
```

```
}
```

```
return false;
```

```
}
```

```
clearTable() {
```

```
    const tableRows = this.tbod.querySelectorAll("tr");
```

```
    tableRows.forEach((row) => {
```

```
        row.innerHTML = "";
```

```
    });
```

```
}
```

```
next() {
```

```
    this.i += 1;
```

```
    this.tbod.classList.add("op");
```

```
    if (new Date(new Date().getFullYear() + 1, new Date().getMonth() - 1, new Date().getDate()) > this.startDate) {
```

```
        if (this.startDate.getMonth() < 11) {
```

```
            this.startDate = new Date(this.startDate.getFullYear(), this.startDate.getMonth() + 1, 1);
```

```
        } else {
```

```
        this.startDate = new Date(this.startDate.getFullYear() + 1, 0, 1);
    }

    this.clearTable();

    this.tarih();
} else {

    alert("En fazla 1 yıl sonrasını görebilirsiniz!");

}


this.tumSip();

this.tak.addEventListener('animationend', () => {

    this.tbod.classList.remove("op");

});

}


prev() {

    this.i -= 1;

    this.tbod.classList.add("op");

    if (new Date() < this.startDate) {

        if (this.startDate.getMonth() > 0) {

            this.startDate = new Date(this.startDate.getFullYear(), this.startDate.getMonth() - 1, 1);

        } else {

            this.startDate = new Date(this.startDate.getFullYear() - 1, 11, 1);

        }

        this.clearTable();

        this.tarih();

    }

    this.tumSip();

    this.tak.addEventListener('animationend', () => {
```

```
    this.tbod.classList.remove("op");  
  });  
}
```

```
tarih() {  
  const options = { weekday: 'long' };  
  
  const ayAdlari = [  
    "Ocak", "Şubat", "Mart", "Nisan", "Mayıs", "Haziran",  
    "Temmuz", "Ağustos", "Eylül", "Ekim", "Kasım", "Aralık"  
  ];  
  
  let month = this.startDate.getMonth()+1; // Bir ay geriye gitmek için -1  
  let year = this.startDate.getFullYear();  
  
  console.log(this.startDate);  
  
  this.ay = ayAdlari[month-1] + " " + year;  
  
  const sira = this.calculateSira(this.startDate.toLocaleDateString('tr-TR', options));  
  
  const daysInMonth = new Date(year, month + 1 , 0).getDate();  
  
  let gun = 1;  
  let spt = 0;  
  let hafta: any = [];
```

```
for (gun; gun <= daysInMonth + sira; gun++) {  
    if (gun <= sira) {  
        hafta.push(null);  
    } else {  
        hafta.push(gun - sira);  
    }  
  
    spt++;  
  
    if (spt === 7) {  
        this.calendar.push(hafta);  
        hafta = [];  
        spt = 0;  
    }  
}  
  
if (spt !== 0) {  
    while (spt < 7) {  
        hafta.push(null);  
        spt++;  
    }  
    this.calendar.push(hafta);  
}
```

```
calculateSira(weekday: string): number {
```

```
    switch (weekday) {
```

```
        case 'Pazartesi':
```

```
            return 0;
```

```
        case 'Salı':
```

```
            return 1;
```

```
        case 'Çarşamba':
```

```
            return 2;
```

```
        case 'Perşembe':
```

```
            return 3;
```

```
        case 'Cuma':
```

```
            return 4;
```

```
        case 'Cumartesi':
```

```
            return 5;
```

```
        case 'Pazar':
```

```
            return 6;
```

```
        default:
```

```
            return 0;
```

```
    }
```

```
}
```

```
tumSip(){
```

```
    console.log(this.bugun,this.startDate);
```

```
    if(this.bugun.getMonth()===this.startDate.getMonth()){
```

```
        this.DateEvent.emit(this.veri);
```

```
        console.log(this.veri);} 
```

```
    else{
```

```
        let bos:any[] =[]
```

```
        this.DateEvent.emit(bos);
```

```

    }

}

selectDate(day: any, month: any, year: any) {
  month+=1;

  if(day<10){
    day="0"+day
  }

  if(month<10){
    month="0"+month
  }

  console.log(`${year}-${month}-${day}T00:00:00`)

  const selectedDate = `${year}-${month}-${day}T00:00:00`;

  this.selectedDateEvent.emit(selectedDate);

}

}

```

---

#### ----product.component.ts----

```

import { Component, OnInit, Renderer2 } from '@angular/core';
import { FormControl, FormGroup, Validators } from '@angular/forms';
import { Route, Router } from '@angular/router';
import { Product_Stock1 } from 'src/app/models/Product';
import { AuthService } from 'src/app/services/auth.service';

```

```

@Component({
  selector: 'app-product',
  templateUrl: './product.component.html',
  styleUrls: ['./../custom_sources/main.css']

```



```

    })

    export class ProductComponent implements OnInit {

        lisance:any;

        pro_form!:Element;
        kat_form!:Element;
        mark_form!:Element;
        blocker!:Element;
        productForm: FormGroup;
        categoryForm:FormGroup;
        markForm:FormGroup;
        veri1:any[]=[];
        veri1copy:any[]=[];
        markalar:any[]=[];
        kategoriler:any[]=[];
        ac_ekran!:Element;
        aciklama!:Element;
        drm=0;
        urun:any;

        constructor(public router:Router,private service:AuthService,private renderer: Renderer2){

            this.productForm = new FormGroup({

                productName: new FormControl("", [Validators.required, Validators.minLength(3),
                Validators.maxLength(50)]),

                category: new FormControl("", Validators.required),

                brand: new FormControl("", Validators.required),

                cost: new FormControl("", [Validators.required, Validators.pattern('[0-9]+$')]),

                sale: new FormControl("", [Validators.required, Validators.pattern('[0-9]+$')]),

                exp: new FormControl ("", [Validators.required, Validators.minLength(3), Validators.maxLength(100)])

            });

```

```

this.categoryForm = new FormGroup({
  katName: new FormControl("", [Validators.required, Validators.minLength(3), Validators.maxLength(50)]),
});

this.markForm = new FormGroup({
  markName: new FormControl("", [Validators.required, Validators.minLength(3), Validators.maxLength(50)]),
});
}

ngOnInit(){
  this.service.getveri().subscribe(
    (data) => {
      this.veri1 = data;
      console.log(this.veri1);
      this.veri1=this.veri1.filter((item)=>item.kategori_ID!=-1);

      this.veri1copy=this.veri1;
      // İstedığınız işlemleri burada gerçekleştirin
    },
    (error) => {
      console.error(error);
      // Hata işlemlerini burada gerçekleştirin
    }
  );

  this.service.getmarka().subscribe(
    (data) => {
      this.markalar = data;
      console.log(this.markalar);
      // İsteddiğiniz işlemleri burada gerçekleştirin
    },
    (error) => {

```

```

        console.error(error);

        // Hata işlemlerini burada gerçekleştirin
    }
);

this.service.getkategori().subscribe(

    (data) => {

        this.kategoriler = data;

        console.log(this.kategoriler);

        // İstedığınız işlemleri burada gerçekleştirin
    },

    (error) => {

        console.error(error);

        // Hata işlemlerini burada gerçekleştirin
    }

);

this.ac_ekran=document.getElementById("ac_ekran");
this.aciklama=document.getElementById("aciklama");
this.pro_form=document.getElementById("pro-add");
this.kat_form=document.getElementById("kat-add");
this.mark_form=document.getElementById("mark-add");
this.blocker=document.getElementById("blocker");
this.alphabetA();

const sk = localStorage.getItem('sk');
const ls = localStorage.getItem('ls');
const mk = localStorage.getItem('mk');
this.lisance=ls;

    if (ls!='2') {

        this.router.navigate(['/hata']);

    } else {

```

```
}  
}
```

```
updateUrun(urun:any){  
  this.pro_ac();  
  
  const urunad=this.productForm.get('productName')!.setValue(urun.urun_ad);  
  const urunkat=this.productForm.get('category')!.setValue(urun.kategori_ID);  
  const urunmaliyet=this.productForm.get('cost')!.setValue(urun.maliyet);  
  const urunsat=this.productForm.get('sale')!.setValue(urun.satis);  
  const urunmark=this.productForm.get('brand')!.setValue(urun.marka_ID);  
  const urunac=this.productForm.get('exp')!.setValue(urun.aciklama);  
  
  this.drm=1;  
  
  this.urun=urun;  
}
```

```
uptadeTamamla(){  
  
  const urunad=this.productForm.get('productName')!.value;  
  const urunkat=parseInt(this.productForm.get('category')!.value);  
  const urunmaliyet=parseInt(this.productForm.get('cost')!.value);  
  const urunsat=parseInt(this.productForm.get('sale')!.value);  
  const urunmark=parseInt(this.productForm.get('brand')!.value);  
  const urunac=this.productForm.get('exp')!.value;  
  
  if(this.urun.urun_ad!=urunad || this.urun.kategori_ID!=urunkat || this.urun.marka_ID!=urunmark  
    ||this.urun.maliyet!=urunmaliyet || this.urun.satis!=urunsat || this.urun.aciklama!=urunac){  
    if(urunmaliyet<urunsat){  
  
this.service.urunupdate(this.urun.urun_ID,urunkat,urunmark,urunsat,urunmaliyet,urunad,urunac).subscribe(  
  () => {
```

```
        console.log('sepet onayı eklendi.');
```

```
        location.reload();
```

```
    },
```

```
    (error) => {
```

```
        console.error('sepet onayı ekleme hatası:', error);
```

```
    }
```

```
);
```

```
location.reload();
```

```
}
```

```
else
```

```
    alert("Ürün maliyeti, ürün satış fiyatından fazla olamaz!");
```

```
}
```

```
else{
```

```
    alert("Aynı değerler girmeyiniz!")
```

```
}
```

```
}
```

```
deleteUrun(urun:any){
```

```
    var sil = confirm(`${urun.urun_ad} ürününü silmek istediğinize emin misiniz?`);
```

```
    if (sil) {
```

```
        this.service.urunsil(parseInt(urun)).subscribe(
```

```
            () => {
```

```
                console.log('sepet onayı eklendi.');
```

```
                location.reload();
```

```
            },
```

```
            (error) => {
```

```
                console.error('sepet onayı ekleme hatası:', error);
```

```

    }

    );

    // location.reload();

} else {

}

}

}

ara(event:any){

    this.veri1copy = this.veri1.filter(item => item.urun_ad.includes(event.target.value));

};

alphabetA(){

    this.veri1.sort(function(a, b) {

        return a.urun_ad.localeCompare(b.urun_ad);

    });

}

alphabetZ(){

    this.veri1.sort(function(a, b) {

        return b.urun_ad.localeCompare(a.urun_ad);

    });

}

insertyeniurun(){

    const urunad=this.productForm.get('productName')!.value;

    const kategoriid=parseInt(this.productForm.get('category')!.value);

    const markaid=parseInt(this.productForm.get('brand')!.value);

    const aciklama=this.productForm.get('exp')!.value;

    const alisfiyat=parseInt(this.productForm.get('cost')!.value);

    const satisfiyat=parseInt(this.productForm.get('sale')!.value);

```

```

const urunvar_m1 = this.veri1.filter((urun)=> urun.urun_ad==urunad && urun.kategori_ID==kategoriid &&
markaid==urun.marka_ID && aciklama==urun.aciklama );

if(urunvar_m1.length>0){
  alert("Aynı üründen var!");
}

else{
  const urunid = this.veri1.sort((a, b) => b.urun_ID - a.urun_ID)[0].urun_ID+1;
  console.log(urunid);
  this.service.insertyeniuurun(urunid,kategoriid, markaid
    , satisfiyat, alisfiyat, urunad,aciklama).subscribe(
    () => {

      console.log('Sipariş eklendi.');
      location.reload();
    },
    (error) => {
      console.error('Sipariş ekleme hatası:', error);
    }
  );
  this.pro_kapat();
  location.reload();
}
}

insertyenikategori(){
  const katad=this.categoryForm.get('katName')!.value;
  const katvar_m1 = this.veri1.filter((urun)=> urun.kategori_AD==katad);
  if(katvar_m1.length>0){
    alert("Aynı kategoriden var!");
  }
}

```

```
else{

this.service.insertkategori(katad).subscribe(

() => {

    console.log('kategori eklendi.');
```

},

```
(error) => {

    console.error('kategori ekleme hatası:', error);

}

);

this.kat_kapat();

location.reload();

}

}
```

```
insertyenimarka(){

const markad=this.markForm.get('markName')!.value;

const markvar_mi = this.veri1.filter((urun)=> urun.marka_AD==markad);

if(markvar_mi.length>0){

    alert("Aynı markadan var!");

}

else{
```

```
this.service.insertmarka(markad).subscribe(

() => {

    console.log('marka eklendi.');
```

},

```
(error) => {

    console.error('marka ekleme hatası:', error);

}

);
```



```
    this.mark_kapat();  
    location.reload();  
  }  
}
```

```
pro_ac(){  
    this.pro_form.classList.remove("d-none");  
    this.blocker.classList.remove("d-none");  
    this.renderer.setStyle(document.body, 'overflow', 'hidden');  
    this.productForm.reset();  
    this.drm=0;  
}
```

```
pro_kapat(){  
    this.pro_form.classList.add("d-none");  
    this.blocker.classList.add("d-none");  
    this.renderer.setStyle(document.body, 'overflow', 'auto');  
    this.productForm.reset();  
}
```

```
kat_ac(){  
    this.kat_form.classList.remove("d-none");  
    this.blocker.classList.remove("d-none");  
    this.renderer.setStyle(document.body, 'overflow', 'hidden');  
    this.categoryForm.reset();  
}
```

```
kat_kapat(){
```

```
this.kat_form.classList.add("d-none");  
  
this.blocker.classList.add("d-none");  
  
this.renderer.setStyle(document.body, 'overflow', 'auto');  
  
this.categoryForm.get('katName')!.reset();  
  
}
```

```
mark_ac(){  
  
    this.mark_form.classList.remove("d-none");  
  
    this.blocker.classList.remove("d-none");  
  
    this.renderer.setStyle(document.body, 'overflow', 'hidden');  
  
    this.markForm.reset();  
  
}
```

```
mark_kapat(){  
  
    this.mark_form.classList.add("d-none");  
  
    this.blocker.classList.add("d-none");  
  
    this.renderer.setStyle(document.body, 'overflow', 'auto');  
  
    this.markForm.get('markName')!.reset();  
  
}
```

```
acikla(acik:any){  
  
    this.ac_ekran.classList.remove("d-none");  
  
    this.blocker.classList.remove("d-none");  
  
    this.renderer.setStyle(document.body, 'overflow', 'hidden');  
  
    this.aciklama.innerHTML=acik.toString();  
  
}
```

```
aciklama_kapat(){  
  
    this.ac_ekran.classList.add("d-none");
```

```
this.blocker.classList.add("d-none");

this.renderer.setStyle(document.body, 'overflow', 'auto');

}
```

```
}
```

#### ----depo.component.ts----

```
import { Component, OnInit } from '@angular/core';
import { AuthService } from '../services/auth.service';
import { Router } from '@angular/router';
```

```
@Component({
  selector: 'app-depo',
  templateUrl: './depo.component.html',
  styleUrls: ['./depo.component.css']
})
```

```
export class DepoComponent implements OnInit {
  constructor(public router : Router, public service : AuthService ) { };
  siparisler:any=[];
  sip:any=[];
  veriler:any=[];
  ustBilesenVeri:any=[];
  ngOnInit(): void {

    if (this.service.getMagazalogin()&&(localStorage.getItem('ls')== '3')) {
      // magazalogin true ise
    } else {
      // magazalogin false ise
      this.router.navigate(['/hata']);
    }
  }
}
```

```
}

this.service.hazirlanan().subscribe(

  (data) => {

    this.siparisler = data;

    this.veriler = data;


    // İstedığınız işlemleri burada gerçekleştirin

  },

  (error) => {

    console.error(error);

    // Hata işlemlerini burada gerçekleştirin

  }

);

this.sip=this.siparisler;

this.veriler=this.sip;

this.ustBilesenVeri = this.sip ;

}

sil(urun:any){

  const siparisadet =parseInt(urun.siparis_adeti); // Siparişin ürün ID'si


  const siparisid =parseInt(urun.siparis_id); // Siparişin ürün ID'si

  //const adet = parseInt( this.adet.value); // Sipariş adeti

  this.service.sepetsil(siparisid,siparisadet).subscribe(

    () => {

      console.log('siparis silindi. ');

      location.reload();

    },

    (error) => {

      console.error('siparis silme hatası:', error);

    }

  )

}
```

```
);  
}
```

```
basaribitir(urun:any){  
  const siparisadet =parseInt(urun.siparis_adeti); // Siparişin ürün ID'si  
  
  const siparisid =parseInt(urun.siparis_id); // Siparişin ürün ID'si  
  //const adet = parseInt( this.adet.value); // Sipariş adeti  
  this.service.basaribitir(siparisid,siparisadet).subscribe(  
    () => {  
      console.log('siparis silindi.');      location.reload();  
    },  
    (error) => {  
      console.error('siparis silme hatası:', error);  
    }  
  );  
}
```

```
handleSelectedDate(selectedDate: string) {  
  console.log(selectedDate);  
  console.log(this.siparisler);  
  this.veriler=this.siparisler.filter((item)=>item.siparis_tarih==selectedDate);  
}
```

```
  Date(veri:any[]){  
    this.veriler=veri;  
  }
```

```

        bitir(urun:any){
            const siparisid =parseInt(urun.siparis_id); // Siparişin ürün ID'si
            const siparisadet =parseInt(urun.siparis_adeti); // Siparişin ürün ID'si
            //const adet = parseInt( this.adet.value); // Sipariş adeti
            this.service.sepetonay(siparisid,siparisadet,3).subscribe(

                () => {

                    console.log('sepet onayı eklendi.');

                    location.reload();

                },

                (error) => {

                    console.error('sepet onayı ekleme hatası:', error);

                }

            );
        }
    }
}

```

#### ----kayit.component.ts----

```

import { Component, OnInit } from '@angular/core';
import { Router } from '@angular/router';
import { Product_Stock1 } from 'src/app/models/Product';
import { AuthService } from 'src/app/services/auth.service';
import { FormsModule } from '@angular/forms';
@Component({
    selector: 'app-kayit',
    templateUrl: './kayit.component.html',
    styleUrls: ['./kayit.component.css']
})
export class KayitComponent implements OnInit {
    urunID:any;
    urunadet:any;

```

```
lisance:any='0';  
adet1: number | undefined;  
blocker !: Element ;  
panel !:Element;  
aciklama !:Element;  
confirm !:any;  
adet !:any;  
drm:number=0;  
satal !:Element;  
urun_yedek:any;  
gercek_stok:any;  
element!:Element;  
    veri1: any[] = [];  
veri1copy: any[] = [];  
kategoriler!:any[];
```

```
constructor(public router : Router,private service:AuthService) {  
  
};  
  
ngOnInit(): void {  
  
// if (this.service.magazalogin==false) {  
    // // this.router.navigate(['/hata']);  
    //  
    // }  
// Örnek kullanım  
if (this.service.getMagazalogin()&&(localStorage.getItem('ls')=='3')) {  
    // magazalogin true ise  
} else {  
    // magazalogin false ise
```

```
this.router.navigate(['/hata']);  
}
```

```
this.service.getkategori().subscribe(  
  (data) => {  
    this.kategoriler = data;  
    console.log(this.kategoriler);  
    // İstedığınız işlemleri burada gerçekleştirin  
  },  
  (error) => {  
    console.error(error);  
    // Hata işlemlerini burada gerçekleştirin  
  }  
);
```

```
this.satal=document.querySelector('#satal')!;  
this.blocker=document.querySelector('#blocker')!;  
this.panel=document.querySelector('#panel')!;  
this.aciklama=document.querySelector('#aciklama')!;  
this.adet=document.querySelector('#adet')!;  
this.confirm=document.querySelector('#confirm')!;  
/*const sk = localStorage.getItem('sk');  
const ls = localStorage.getItem('ls');  
this.lisance=ls;  
if (ls!='4') {  
  this.router.navigate(['/hata']);  
  
} else {
```



```
    }  
    */
```

```
this.alphabetA();  
  
const sk = localStorage.getItem('sk');  
  
const ls = localStorage.getItem('ls');  
  
const mk = localStorage.getItem('mk');  
  
this.lisance=ls;
```

```
this.service.getveri().subscribe(  
  
  (data) => {  
  
    this.veri1 = data;  
  
    this.veri1=this.veri1.filter((item)=>item.kategori_ID!=-1);  
  
    console.log(this.veri1);  
this.veri1copy = this.veri1;
```

```
    // İstedığınız işlemleri burada gerçekleştirin  
  
  },  
  
  (error) => {  
  
    console.error(error);  
  
    // Hata işlemlerini burada gerçekleştirin  
  
  }  
);
```

```
const sepet= localStorage.getItem('sepet');  
  
const siparis= localStorage.getItem('siparis');  
  
}
```

```
accc(urun : Product_Stock1) {  
  if(this.drm==1){  
    this.aciklama.innerHTML=urun.urun_ad+" "+urun.aciklama;  
    this.aciklama.classList.remove("d-none");  
    this.drm=0;  
  }  
  this.panel.classList.remove("d-none");  
  this.blocker.classList.remove('d-none');  
  document.body.style.overflow="hidden";  
}
```

```
adetekle(urun:any,adet:number){  
  this.satal.classList.remove("d-none");  
  this.panel.classList.remove("d-none");  
  this.blocker.classList.remove('d-none');  
  document.body.style.overflow="hidden";  
  this.urunID =parseInt(urun.urun_ID);  
  this.urunadet =parseInt(urun.urun_adet);  
}
```

```
ara(event:any){  
  this.veri1copy = this.veri1.filter(item => item.urun_ad.includes(event.target.value));  
};
```

```
tamamla(adet:any) {  
  
  this.adet1 =parseInt(adet);  
  this.adet1=(this.urunadet+this.adet1);  
  
  console.log(this.urunID,adet);  
  if (adet.value>0)  
    this.service.adetekle(this.urunID, this.adet1!).subscribe(  
      () => {  
        console.log('urun adeti eklendi.');        this.adet.value="";  
      }  
    );  
  location.reload();  
}
```

```
  },  
  (error) => {  
    console.error('urun adeti ekleme hatası:', error);  
  }  
);
```

```
  this.kapt();
```

```
}
```

```
alphabetA(){  
  this.veri1.sort(function(a, b) {  
    return a.urun_ad.localeCompare(b.urun_ad);  
  });  
}  
alphabetZ(){
```

```
this.veri1.sort(function(a, b) {  
    return b.urun_ad.localeCompare(a.urun_ad);  
});  
}
```

```
kapt(){  
  
    this.panel.classList.add('d-none');  
    this.blocker.classList.add('d-none');  
    this.aciklama.classList.add("d-none");  
    this.satal.classList.add("d-none");  
    document.body.style.overflow="auto";  
    this.adet.value="";  
}  
}
```

---

#### ----gecmis.component.ts----

```
import { Component } from '@angular/core';  
//import { AuthService } from 'src/app/services/auth.service';  
import { Router } from '@angular/router';  
import { AuthService } from 'src/app/services/auth.service';  
@Component({  
    selector: 'app-gecmis',  
    templateUrl: './gecmis.component.html',  
    styleUrls: ['./gecmis.component.css']  
})  
export class GecmisComponent {
```

```

    lisance:any='0';

    gecmis:any[]=[];

    constructor(public router : Router, public service : AuthService ) {

    };

    ngOnInit(): void {

        if (this.service.getMagazalogin()) {

            // magazalogin true ise

        } else {

            // magazalogin false ise

            this.router.navigate(['/hata']);

        }

        // if (this.service.magazalogin==false) {

        // // this.router.navigate(['/hata']);

        //

        // }

    }

    this.service.gecmis().subscribe(

        (data) => {

            this.gecmis = data;

            // console.log(this.gecmis);

            // İstedığınız işlemleri burada gerçekleştirin

        },

        (error) => {

            console.error(error);

            // Hata işlemlerini burada gerçekleştirin

        }

    );

}

```

```
}
```

-----  
**----magaza.component.ts----**

```
import { Component, Input, OnInit } from '@angular/core';
import { Router } from '@angular/router';
//import { AuthService } from 'src/app/services/auth.service';
import { Product_Stock1 } from "src/app/models/Product";
import { SepetComponent } from './sepet/sepet.component';
import { AuthService } from 'src/app/services/auth.service';

@Component({
  selector: 'app-magaza',
  templateUrl: './magaza.component.html',
  styleUrls: ['./magaza.component.css']
})
export class MagazaComponent implements OnInit {

  lisance:any='0';


  blocker !: Element ;
  panel !:Element;
  aciklama !:Element;
  confirm !:Element;
  adet !:any;
  drm:number=0;
  satal !:Element;
  urun_yedek:any;
  gercek_stok:any;
  element!:Element;

  veri1: any[] = [];
  kategoriler:any[]=[];
  veri1copy: any[]=[];
```

```
constructor(public router : Router,private service:AuthService) {  
  
};  
  
  ngOnInit(): void {  
  
  
  
  
  
  
    // if (this.service.magazalogin==false) {  
      // // this.router.navigate(['/hata']);  
  
      //  
      // }  
    // Örnek kullanım  
    if (this.service.getMagazalogin()&&(localStorage.getItem('ls')=='4')) {  
      // magazalogin true ise  
  
    } else {  
      // magazalogin false ise  
  
      this.router.navigate(['/hata']);  
  
    }  
  

```

```
this.service.getkategori().subscribe(  
  
  (data) => {  
  
    this.kategoriler = data;  
  
    // console.log(this.kategoriler);  
  
    // İstedığınız işlemleri burada gerçekleştirin  
  
  },  
  
  (error) => {  
  
    console.error(error);  
  
    // Hata işlemlerini burada gerçekleştirin  
  
  }  
  
);
```

```
this.satal=document.querySelector('#satal')!;  
this.blocker=document.querySelector('#blocker')!;  
this.panel=document.querySelector('#panel')!;  
this.aciklama=document.querySelector('#aciklama')!;  
this.adet=document.querySelector('#adet')!;  
this.confirm=document.querySelector('#confirm')!;
```

```
this.alphabetA();  
const sk = localStorage.getItem('sk');  
const ls = localStorage.getItem('ls');  
const mk = localStorage.getItem('mk');  
this.lisance=ls;  
if (ls!='4') {  
  // this.router.navigate(['/hata']);  
  
} else {  
}
```

```
this.service.getveri().subscribe(  
  (data) => {  
    this.veri1 = data;  
    this.veri1=this.veri1.filter((item)=>item.kategori_ID!=-1);  
    this.veri1copy=this.veri1;  
  
    //console.log(this.veri1);  
    // İstedığınız işlemleri burada gerçekleştirin
```



```
},  
  
(error) => {  
  
  // console.error(error);  
  
  // Hata işlemlerini burada gerçekleştirin  
  
}  
  
);
```

```
const sepet= localStorage.getItem('sepet');  
const siparis= localStorage.getItem('siparis');  
  
}
```

```
ara(event:any){  
  
  this.veri1copy = this.veri1.filter(item => item.urun_ad.includes(event.target.value));  
  
};
```

```
accc(urun : Product_Stock1) {  
  
if(this.drm==1){  
  
  this.aciklama.innerHTML=urun.urun_ad+" "+urun.aciklama;  
  
  this.aciklama.classList.remove("d-none");  
  
  this.drm=0;  
  
}
```

```
this.panel.classList.remove("d-none");  
this.blocker.classList.remove('d-none');  
document.body.style.overflow="hidden";  
}
```

```
satinal(urun : Product_Stock1,event :any){  
if(event.target.textContent=="Sepete git")  
{  
this.router.navigate(['/magaza/sepet']);  
}  
}
```

```
this.satal.classList.remove("d-none");  
this.panel.classList.remove("d-none");  
this.blocker.classList.remove('d-none');  
document.body.style.overflow="hidden";
```

```
this.urun_yedek=urun;  
this.gercek_stok=urun.iss;  
this.adet.value="";  
}
```

```
tamamla() {
```

```

    if (this.gercek_stok >= this.adet.value) {

const urunid =parseInt( this.urun_yedek.urun_ID); // Siparişin ürün ID'si

const adet = parseInt( this.adet.value); // Sipariş adeti

this.service.insertSiparis(urunid, adet).subscribe(

    () => {

        // console.log('Sipariş eklendi.');
```

},

```

        (error) => {

            console.error('Sipariş ekleme hatası:', error);

        }
    });

    this.urun_yedek.stok = this.adet.value;

    this.urun_yedek.drm=5;

    // localStorage.setItem(this.urun_yedek.id, JSON.stringify(this.urun_yedek));

    this.urun_yedek.stok = this.gercek_stok;

} else {

    alert("Stokdaki üründen fazla ürün talep edemezsiniz!");

    return;

}

this.urun_yedek = null;

this.kapt();

}

```

```

alphabetA(){

    this.veri1.sort(function(a, b) {

```

```

        return a.urun_ad.localeCompare(b.urun_ad);
    });
}
alphabetZ(){
    this.veri1.sort(function(a, b) {
        return b.urun_ad.localeCompare(a.urun_ad);
    });
}

```

```

kapt(){
    this.panel.classList.add('d-none');
    this.blocker.classList.add('d-none');
    this.aciklama.classList.add("d-none");
    this.satal.classList.add("d-none");
    document.body.style.overflow="auto";
}
}

```

#### ----sepet.component.ts----

```

import { Component, Input } from '@angular/core';
//import { AuthService } from 'src/app/services/auth.service';
import { Router } from '@angular/router';
import { elements } from 'chart.js';
import { Product_Stock1 } from 'src/app/models/Product';
import { Product_Depo } from 'src/app/models/Products_Depo';
import { AuthService } from 'src/app/services/auth.service';

```

```

@Component({
    selector: 'app-sepet',

```

```

    templateUrl: './sepet.component.html',
    styleUrls: ['./sepet.component.css']
  })
export class SepetComponent {

  element:Element | undefined;

  sepet:any[]=[];

  lisance:any='0';


  Product_Depo : Product_Depo;

  constructor(public router : Router,public service:AuthService) {

    this.Product_Depo=new Product_Depo();
  };

  ngOnInit(): void {

    if (this.service.getMagazallogin()) {

      // magazallogin true ise

    } else {

      // magazallogin false ise

      this.router.navigate(['/hata']);

    }

    // if (this.service.magazallogin==false) {

    // // this.router.navigate(['/hata']);

    //

    // }

    this.service.sepet().subscribe(

      (data) => {

        this.sepet = data;

        // console.log(this.sepet);

```

```
    // İsteddiğiniz işlemleri burada gerçekleştirin

  },

  (error) => {

    console.error(error);

    // Hata işlemlerini burada gerçekleştirin

  }

);
```

```
this.yenile();
```

```
    /*const sk = localStorage.getItem('sk');

    const ls = localStorage.getItem('ls');

    this.lisance=ls;

    if (ls!='4') {

      this.router.navigate(['/hata']);

    } else {

    }

    */

  }
```

```
yenile(){
```

```
    /*fonksiyon drm 5 gelenler authservice.sepette.subscribe(src => data)
```

```
*/
```

```
}
```

```
hepsiniAl() {
```

```
  for (let urun of this.sepet) {
```

```
    if (urun.iss>=urun.siparis_adeti) {
```

```
      this.satinal(urun);
```

```
    }
```

```
  else
```

```
    alert(urun.urun_ad+" adlı ürün yeterli stoğa sahip değil");
```

```
}
```

```
}
```

```
sil(urun:any){
```

```
  const siparisadet =parseInt(urun.siparis_adeti); // Siparişin ürün ID'si
```

```
  const siparisid =parseInt(urun.siparis_id); // Siparişin ürün ID'si
```

```
//const adet = parseInt( this.adet.value); // Sipariş adeti
```

```
this.service.sepetsil(siparisid,siparisadet).subscribe(
```

```
  () => {
```

```
    // console.log('sepet silindi.');
```

```
    location.reload();
```

```
  },
```

```
(error) => {
```

```
  console.error('sepetsilinme hatası:', error);
```

```
}
```

```
);
```

```
}
```

```
hepsinisi(){

}

//şuraya gety ile users çekildi sirket_id bilgisi var sen bunu
satinal(urun:any){

    const siparisid =parseInt(urun.siparis_id); // Siparişin ürün ID'si
    const siparisadet =parseInt(urun.siparis_adeti); // Siparişin ürün ID'si
    //const adet = parseInt( this.adet.value); // Sipariş adeti
    this.service.sepetonay(siparisid,siparisadet,1).subscribe(

        () => {

            // console.log('sepet onayı eklendi. ');

            location.reload();

        },

        (error) => {

            console.error('sepet onayı ekleme hatası:', error);

        }

    );

}

kontrol(event: any,urun : any){

    var deger = event.target.value;

    var gerçek_stok=urun.urun_adet;

    if(deger<=0 || deger===null || deger>gerçek_stok){

        alert("adet boş veya 0 veya stoktan fazla girilemez!");

        event.target.value=urun.siparis_adeti;

    }

}
```



```

    }

    else{

urun.siparis_adeti=event.target.value;


const siparisid =parseInt(urun.siparis_id); // Siparişin ürün ID'si
const siparisadet =parseInt(urun.siparis_adeti); // Siparişin ürün ID'si
//const adet = parseInt( this.adet.value); // Sipariş adeti
this.service.sepetonay(siparisid,siparisadet,0).subscribe(

    () => {

// console.log('sepet onayı eklendi. ');

location.reload();

    },

    (error) => {

console.error('sepet onayı ekleme hatası:', error);

    }

);

}

}

}

```

#### -----siparisler.component.ts-----

```

import { Component } from '@angular/core';

//import { AuthService } from 'src/app/services/auth.service';

import { Router } from '@angular/router';

import { AuthService } from 'src/app/services/auth.service';

@Component({

    selector: 'app-siparisler',

    templateUrl: './siparisler.component.html',

    styleUrls: ['./siparisler.component.css']

})

```

```
export class SiparislerComponent {

  lisance:any='0';

  siparisler:any[][]=[];

  constructor(public router : Router,public service:AuthService) {

  };

  ngOnInit(): void {

    if (this.service.getMagazalogin()) {

      // magazalogin true ise

    } else {

      // magazalogin false ise

      this.router.navigate(['/hata']);

    }

  }

  this.service.getsiparislerim().subscribe(

    (data) => {

      this.siparisler = data;

      // console.log(this.siparisler);

      // İstedığınız işlemleri burada gerçekleştirin

    },

    (error) => {

      console.error(error);

      // Hata işlemlerini burada gerçekleştirin

    }

  );

}

sil(urun:any){

  const siparisadet =parseInt(urun.siparis_adeti); // Siparişin ürün ID'si
```

```

const siparisid =parseInt(urun.siparis_id); // Siparişin ürün ID'si

//const adet = parseInt( this.adet.value); // Sipariş adeti

this.service.sepetsil(siparisid,siparisadet).subscribe(

  () => {

    // console.log('sepet silindi. ');

    location.reload();

  },

  (error) => {

    console.error('sepetsilinme hatası:', error);

  } );

}

}

```

#### ----navbar.component.ts----

```

import { Component, OnInit,ElementRef, Renderer2, HostListener } from '@angular/core';
import { FormControl, FormGroup, Validators } from '@angular/forms';
import { ActivatedRoute, Router } from '@angular/router';
import { AuthService } from 'src/app/services/auth.service';
import{Product_Stock1} from 'src/app/models/Product'
import{Product_Depo} from 'src/app/models/Products_Depo'

```

```

@Component({
  selector:"navbar",
  templateUrl: "../navbar.component.html",
  styleUrls: ["../custom_sources/main.css"]
})

export class NavbarComponent implements OnInit {

  message: string = 'asd';

  mesaj:string="";

  sayac:number=0;

```

```
private blocker  !: Element;
private loginn   !: Element;
private cross    !: Element;
private log_div  !: Element;
private log_button!: Element;
private gir_btn  !: Element;
private kayit_btn !: Element;
private kayit_tık !: Element;
private log_btn  !: Element;
private username !: any;
private giris :any;
private reg !:Element;
private log_divi !:Element;
logCnt="";
main=false;
private sayac2:any;
```

```
constructor(private loginAuth: AuthService,public router: Router,private elementRef: ElementRef, private
render: Renderer2) {
```

```
}
```

```
ngOnInit() {
```

```
if (localStorage.getItem("sayac2")== '1') {
  this.startIdleTimer = this.startIdleTimer.bind(this);
  this.resetIdleTime = this.resetIdleTime.bind(this);
  this.checkIdleTime = this.checkIdleTime.bind(this);
  this.giris=new Date(localStorage.getItem('giris')!)
  this.startIdleTimer();
```

```
}
```

```
this.gir_btn = document.getElementById('actLOG');  
this.log_button =document.querySelector('#logButton');  
this.log_div =document.querySelector('#log');  
this.log_divi =document.querySelector('#logi');  
this.cross = document.querySelector('#cross');  
this.blocker = document.querySelector('#blocker');  
this.loginn = document.querySelector('#login');  
this.kayit_btn = document.querySelector('#actREG');  
this.kayit_tik = document.querySelector('#kayit_tik');  
this.log_btn = document.querySelector('#loginCont');  
this.username = document.querySelector('#username');  
this.reg=document.querySelector('#reg');
```

```
setTimeout(() => {  
  if (localStorage.getItem("cikis")== '1') {  
    localStorage.setItem("cikis", '0');  
    alert('60 dakika boyunca etkileşimde bulunmadınız. Bu yüzden otomatik çıkış yapılmıştır.');
```

```
}
```

```
}, 311);
```

```
}
```

```
/*
```

```
@HostListener('window:beforeunload', ['$event'])
confirmExit(event: BeforeUnloadEvent) {
  if(this.router.url!==""){
    event.preventDefault();
    event.returnValue = 'Emin misiniz?';
  }
  this.logout();
} */
```

```
async loginSubmitted() {
  try {
    const res = await this.loginAuth.loginruser([this.email.value, this.pass.value]).toPromise();

    if (res === 'Failure') {
      alert('Login Unsuccessful');
    } else {

      this.sayac2=1;
      localStorage.setItem("sayac2",this.sayac2);
      // this.loginAuth.magazallogin=true;
      // Örnek kullanım
      localStorage.setItem('email',this.email.value);
      this.loginAuth.setMagazallogin(true);
      localStorage.setItem('giris',new Date().toString());

      const data = JSON.parse(res!);
      const veri = data.veri;
```

```
const sepet = data.sepet;

const siparis = data.siparis;

const mk = data.mk;

localStorage.setItem('mk1',data.mk);

localStorage.setItem('id',data.id);

this.loginAuth.updateAPIUrl(data.sk);

this.loginAuth.siparis1 = siparis;


if (data.ls === '4') {

    localStorage.setItem('ls',data.ls);

    this.router.navigate(['/magaza']);

} else if (data.ls === '3') {

    localStorage.setItem('ls',data.ls);

    this.router.navigate(['/depo']);

} else if (data.ls === '2') {

    localStorage.setItem('ls',data.ls);

    this.router.navigate(['/islemci']);

} else if (data.ls === '1') {

    localStorage.setItem('ls',data.ls);

    this.router.navigate(['/main']);

}


this.kapat();

this.storeDataInLocalStorage(veri, sepet, siparis, mk);


}

} catch (error) {

    console.error('An error occurred during login:', error);

}

}

idleTime = 0;
```

```
idleTimeout = 50 * 60 * 1000; // 50 dakika (dakika * saniye * milisaniye)
```

```
logoutTimeout = 60 * 60 * 1000; // 1 saat (dakika * saniye * milisaniye)
```

```
idleInterval:any;
```

```
startIdleTimer() {
```

```
    this.idleInterval = setInterval(this.checkIdleTime, 60000); // Her dakikada bir zamanlayıcıyı kontrol et
```

```
    this.resetIdleTime();
```

```
    console.log("çalıştı")
```

```
    document.addEventListener('mousemove', this.resetIdleTime); // Fare hareket ettirildiğinde  
    zamanlayıcıyı sıfırla
```

```
    document.addEventListener('keypress', this.resetIdleTime); // Tuşa basıldığında zamanlayıcıyı sıfırla
```

```
}
```

```
resetIdleTime() {
```

```
    this.idleTime = 0;
```

```
}
```

```
checkIdleTime() {
```

```
    this.idleTime += 60000; // Her seferinde 1 dakika (dakika * saniye * milisaniye)
```

```
    console.log(this.idleTime);
```

```
    if (this.idleTime === this.idleTimeout) {
```

```
        // Kullanıcıya uyarı mesajı ver
```

```
    } else if (this.idleTime === this.logoutTimeout) {
```

```
        // Kullanıcının oturumunu sonlandır
```

```
        let dakika=(((new Date().getHours()*60)+(new Date().getMinutes()))-60)-  
        ((this.giris.getHours()*60)+(this.giris.getMinutes()));
```

```
        this.loginAuth.dakikaekle(dakika).subscribe(
```

```
            () => {
```

```
                console.log('Dakika güncellendi.');
```

```
            },
```

```
            (error) => {
```

```
                console.error('Dakika güncelleme hatası:', error);
```



```
    }  
  );  
  console.log(dakika);  
  this.logoutUser();  
  this.logout2();  
  localStorage.setItem("cikis",'1');
```

```
  }  
}
```

```
logoutUser() {  
  clearInterval(this.idleInterval);  
  // Zamanlayıcıyı temizle  
}
```

```
storeDataInLocalStorage(veri: any, sepet: any, siparis: any, mk: any) {  
  localStorage.setItem('veri', JSON.stringify(veri));  
  localStorage.setItem('sepet', JSON.stringify(sepet));  
  localStorage.setItem('siparis', JSON.stringify(siparis));  
  localStorage.setItem('mk', JSON.stringify(mk));  
  console.log(localStorage.getItem('mk'));  
}
```

```
loginForm = new FormGroup({  
  email: new FormControl("", [Validators.required, Validators.email]),
```

```

    pass: new FormControl("", [Validators.required, Validators.minLength(6), Validators.maxLength(15)]),
  });

  logout() {

    this.sayac2=0;

    let dakika=(((new Date().getHours()*60)+(new Date().getMinutes())))-
    ((this.giris.getHours()*60)+(this.giris.getMinutes()));

    this.loginAuth.dakikaekle(dakika).subscribe(

      () => {

        console.log('Dakika güncellendi.');
```

},

```

      (error) => {

        console.error('Dakika güncelleme hatası:', error);

      }

    );

    this.logoutUser();

    // this.loginAuth.magazallogin=false;
    this.loginAuth.setMagazallogin(false);

  }

  this.loginAuth.logout();

  this.router.navigate(['/']);

  this.ac();
}

get email(): FormControl {

  return this.loginForm.get('email') as FormControl;

}

get pass(): FormControl {

  return this.loginForm.get('pass') as FormControl;

}

//accounts = [{username : "abc",password : "123"},{username : "def",password : "456"}];
```

```
handleClick(event: Event) {  
    event.preventDefault();  
}  
  
logout2(){  
    this.loginAuth.logout();  
    this.router.navigate(['/']);  
    this.ac();  
  
}
```

```
kapat(){  
    this.renderer.removeClass(this.blocker!, 'd-block');  
    this.renderer.addClass(this.blocker!, 'd-none');  
    this.renderer.removeClass(this.loginn!, 'd-block');  
    this.renderer.addClass(this.loginn!, 'd-none');  
    this.renderer.setStyle(document.body, 'overflow', 'auto');  
}
```

```
ac() {  
    this.renderer.removeClass(this.blocker!, 'd-none');  
    this.renderer.addClass(this.blocker!, 'd-block');  
    this.renderer.removeClass(this.loginn!, 'd-none');  
    this.renderer.addClass(this.loginn!, 'd-block');
```

```
    this.renderer.setStyle(document.body, 'overflow', 'hidden');  
  }
```

```
gir(){  
  this.renderer.removeClass(this.kayit_btn!, 'act');  
  this.renderer.addClass(this.gir_btn!, 'act');  
  this.renderer.addClass(this.log_div!, 'op');  
  
  this.renderer.removeClass(this.log_divi!, 'd-none');  
  this.renderer.addClass(this.reg!, 'd-none');
```

```
  this.log_div.addEventListener('animationend', ()=>{  
    this.log_div.classList.remove("op");  
  });  
}
```

```
kayit(){  
  
  this.gir_btn.classList.remove("act");  
  this.kayit_btn.classList.add("act");  
  this.log_div.classList.add("op");  
  
  this.renderer.removeClass(this.reg!, 'd-none');  
  this.renderer.addClass(this.log_divi!, 'd-none');
```

```
this.log_div.addEventListener('animationend',()=>{  
    this.log_div.classList.remove("op");  
});  
  
}  
  
}
```

---

#### ----yonet-hesap.component.ts----

```
import { Component, Renderer2 } from '@angular/core';  
import { FormControl, FormGroup, Validators } from '@angular/forms';  
import { Router, RouterLink } from '@angular/router';  
import { AuthService } from 'src/app/services/auth.service';  
import { HostListener } from '@angular/core';
```

```
@Component({  
    selector: 'app-yonet-hesap',  
    templateUrl: './yonet-hesap.component.html',  
    styleUrls: ['./yonet-hesap.component.css']  
})
```

```
export class YonetHesapComponent {  
    personels: any[] = [];  
    roller: any[] = [];  
    rol_ad = "";  
    eski_email:any;  
    personelForm: FormGroup;  
    drm = 0;  
    pro_form!: Element;  
    blocker!: Element;  
    personel: any;
```

```

    girislerCopy: any[]=[];

    girisler: any[]=[];

    constructor(public router: Router, private renderer: Renderer2, public service: AuthService) {

        this.personelForm = new FormGroup({

            personel_email: new FormControl("", [Validators.required, Validators.pattern('^\\w+([.-]?\\w+)*@\\w+([.-]?\\w+)*(.\\w{2,3})+$')]),

            personel_sifre: new FormControl("",[Validators.required, Validators.minLength(6)]),

            personel_rol: new FormControl("", Validators.required),

            personel_isim: new FormControl("",[Validators.required, Validators.pattern('^[a-zA-ZğüşöçİĞÜŞÖÇ]+$')])

        });

    }

    personelsay: any;

    ngOnInit() {

        this.service.getdakika().subscribe(

            (data) => {

                this.girisler = data;

                this.girislerCopy = data;

                // İstedığınız işlemleri burada gerçekleştirin

            },

            (error) => {

                console.error(error);

                // Hata işlemlerini burada gerçekleştirin

            }

        );

    }

    if ((localStorage.getItem('ls')==1)) {

        // magazalogin true ise

    } else {

        // magazalogin false ise

        this.router.navigate(['/hata']);
    }

```

```
}  
  
console.log(localStorage.getItem('id'));  
  
this.service.getuser(parseInt(localStorage.getItem('mk1'))).subscribe(  
  (data) => {  
    this.personels = data;  
    this.personelsay = this.personels.length;  
    console.log(this.personels);  
    // İstedığınız işlemleri burada gerçekleştirin  
  },  
  (error) => {  
    console.error(error);  
    // Hata işlemlerini burada gerçekleştirin  
  }  
);  
  
this.service.getlisance().subscribe(  
  (data) => {  
    this.roller = data;  
    console.log(this.roller);  
    // İsteddiğiniz işlemleri burada gerçekleştirin  
  },  
  (error) => {  
    console.error(error);  
    // Hata işlemlerini burada gerçekleştirin  
  }  
);  
  
this.pro_form = document.getElementById("pro-add");  
this.blocker = document.getElementById("blocker");
```

```
}
```

```
duzenle(personel: any) {  
    this.pan_ac();  
    const personelemail = this.personelForm.get('personel_email')!.setValue(personel.email);  
    const personelsifre = this.personelForm.get('personel_sifre')!.setValue(personel.pass);  
    const personelrol = this.personelForm.get('personel_rol')!.setValue(personel.ls);  
    const personelisim = this.personelForm.get('personel_isim')!.setValue(personel.firstLast);  
    this.drm = 1;  
    this.eski_email=this.personelForm.get('personel_email')!.value;  
    if (this.personelForm.get('personel_rol')!.value == 1) {  
        alert("Şifre yada E-mailiniz değiştirerseniz değişiklikten sonra çıkış yapacaktır!");  
        document.querySelector('#pro-kat')!.classList.add('d-none');  
    } else {  
        document.querySelector('#pro-kat')!.classList.remove('d-none');  
    }  
    this.personel = personel;  
}  
  
sil(personel: any) {  
    var sil = confirm(`${personel.firstLast} adlı personeli silmek istediğinize emin misiniz? \n \n Eğer Silerseniz: \n 1.Personellerden personel kaldırılacak.`);  
  
    if (sil) {  
        //silme api fonksiyonu  
        console.log(personel.id);
```



```

this.service.usersil( parseInt(personel.id)).subscribe(
  (error) => {
    console.error(error);
    // Hata işlemlerini burada gerçekleştirin
  }
);
location.reload();
} else {

}
}
hesap(per:any){
  this.girislerCopy=this.girisler.filter((item)=>item.id==per.id);
}

hepsi(){
  this.girislerCopy=this.girisler.filter((item)=>item);
}
uptadeTamamla() {
  const personelemail = this.personelForm.get('personel_email')!.value;
  const personelsifre = this.personelForm.get('personel_sifre')!.value;
  const personelrol = this.personelForm.get('personel_rol')!.value;
  const personelisim = this.personelForm.get('personel_isim')!.value;

  if (this.personel.email != personelemail || this.personel.pass != personelsifre || this.personel.ls !=
personelrol
    || this.personel.firstLast != personelisim) {

this.service.updateuser(personelisim,personelemail,parseInt(personelrol),this.eski_email,personelsifre,pars
eInt(localStorage.getItem('mk1')!)).subscribe(
  () => {

```

```
        console.log('hesap onayı eklendi.');
```

```
        location.reload();
```

```
    },
```

```
    (error) => {
```

```
console.log(personelisim,personelemail,parseInt(personelrol),personelemail,personelsifre,parseInt(localStorage.getItem('mk1')));
```

```
        console.error('hesap onayı ekleme hatası:', error);
```

```
    }
```

```
);
```

```
if (personelrol.value == 1)
```

```
    if (this.personel.email != personelemail || this.personel.pass != personelsifre) {
```

```
        this.service.logout();
```

```
    }
```

```
    else {
```

```
        location.reload();
```

```
    }
```

```
}
```

```
else {
```

```
    alert("Aynı değerler girmeyiniz!")
```

```
}
```

```
}
```

```
insertyenipersonel() {
```

```

const personelemail = this.personelForm.get('personel_email')!.value;
const personelsifre = this.personelForm.get('personel_sifre')!.value;
const personelrol = this.personelForm.get('personel_rol')!.value;
const personelisim = this.personelForm.get('personel_isim')!.value;

const personelvar_mi = this.personels.filter((per) => per.email == personelemail && per.pass ==
personelsifre && per.firstLast == personelisim && per.ls == personelrol);

if (personelvar_mi.length > 0) {
    alert("Personel var!");
}
else {
    // firstlast:string,email1:string,ls1:number, pass1:string, sirket_id: number

this.service.insertyeniuye(personelisim,personelemail,parseInt(personelrol),personelsifre,parseInt(localStorage.getItem('mk1')!)).subscribe(
    () => {
        console.log('sepet onayı eklendi.');
```

```

console.log(personelisim,personelemail,parseInt(personelrol),personelsifre,parseInt(localStorage.getItem('mk1')!));

        location.reload();
    },
    (error) => {
        console.error('sepet onayı ekleme hatası:', error);

console.log(personelisim,personelemail,parseInt(personelrol),personelsifre,parseInt(localStorage.getItem('mk1')!));

    }
);

//api fonksiyon üstteki değerlerden çek
this.pan_kapat();

```

```
// location.reload();  
  
}  
  
}
```

```
pan_ac() {  
  
  document.querySelector('#pro-kat')!.classList.remove('d-none');  
  
  this.pro_form.classList.remove("d-none");  
  
  this.blocker.classList.remove("d-none");  
  
  this.renderer.setStyle(document.body, 'overflow', 'hidden');  
  
  this.personelForm.reset();  
  
  this.drm = 0;  
  
}
```

```
pan_kapat() {  
  
  this.pro_form.classList.add("d-none");  
  
  this.blocker.classList.add("d-none");  
  
  this.renderer.setStyle(document.body, 'overflow', 'auto');  
  
  this.personelForm.reset();  
  
}  
  
}
```

-----

### ----islemci.component.ts----

```
import { Component, OnInit } from '@angular/core';  
  
import { Router,RouterLink } from '@angular/router';  
  
import { AuthService } from 'src/app/services/auth.service';
```

```
@Component({
```

```
selector: 'app-islemci',

templateUrl: './islemci.component.html',

styleUrls: ['./islemci.component.css']

})

export class IslemciComponent implements OnInit {

  constructor(public router: Router, public service: AuthService) { }

  islemcisepet:any=[];

  ngOnInit(): void {

    if (this.service.getMagazalogin() && (localStorage.getItem('ls') === '2')) {

      // magazalogin true ise


      this.service.islemcisepet().subscribe(

        (data) => {

          this.islemcisepet = data;

          console.log(this.islemcisepet);

          // İstedığınız işlemleri burada gerçekleştirin

        },

        (error) => {

          console.error(error);

          // Hata işlemlerini burada gerçekleştirin

        }

      );

    } else {

      // magazalogin false ise
```

```
this.router.navigate(['/hata']);  
  
}  
  
}  
  
satinal(urun:any){  
  
    const siparisid =parseInt(urun.siparis_id); // Siparişin ürün ID'si  
    const siparisadet =parseInt(urun.siparis_adeti); // Siparişin ürün ID'si  
  
    this.service.sepetonay(siparisid,siparisadet,2).subscribe(  
        () => {  
            console.log('islem hazırlanıyor.');            location.reload();  
        },  
        (error) => {  
            console.error('işlem hatası:',urun, error);  
        }  
    );  
}  
  
sil(urun:any){  
    const siparisadet =parseInt(urun.siparis_adeti); // Siparişin ürün ID'si  
  
    const siparisid =parseInt(urun.siparis_id); // Siparişin ürün ID'si  
    //const adet = parseInt( this.adet.value); // Sipariş adeti  
    this.service.sepetsil(siparisid,siparisadet).subscribe(  
        () => {  
            console.log('sepet silindi.');        }  
    );  
}
```

```

        location.reload();

    },

    (error) => {

        console.error('sepetisilinme hatası:', error);

    }

);

}

}

```

## 8.CSS Kodları

----main.css----

```

@import
url("https://fonts.googleapis.com/css2?family=Montserrat:ital,wght@0,100;0,200;0,300;0,400;0,500;0,600;0,800;0,900;1,100;1,200;1,300;1,400;1,500;1,600;1,700;1,800;1,900&display=swap");.yesil{background-color:#07b332 !important;border:1px solid white !important}.kirmizi{background-color:#a32020 !important;border:1px solid white !important}.sari{background-color:#c0a80f !important;border:1px solid white !important}.guzel1{color:#a8214a}.guzel2{color:#288a07}.hei{height:400px !important}.hei-xl{height:calc(800px + 0.5rem)}.hei-md{height:calc(550px)}.hei-small{height:250px}.cont-bg{background-color:white !important}.grey{background:rgba(239,238,238,0.154) !important}::-webkit-scrollbar{width:15px}.font{font-size:16px !important}.font h6{font-size:17px !important}.font h5{font-size:20px !important}.font h4{font-size:25px !important}.bg{background:#1236d9 !important;background:linear-gradient(90deg, #1236d9 0%, #0684ec 49%, #27b4d7 100%) !important}html,body{position:relative;background:#1236d9 !important;background:linear-gradient(90deg, #1236d9 0%, #0684ec 49%, #27b4d7 100%) !important}::-webkit-scrollbar-track{background-color:transparent}::-webkit-scrollbar-thumb{background-color:#ffffff;border-radius:15px;border:6px solid transparent;background-clip:content-box}::-webkit-scrollbar-thumb:hover{background-color:#c9c9c9}html ::-webkit-scrollbar{width:15px}html ::-webkit-scrollbar-thumb{background-color:#1d4b79;background-clip:content-box}html ::-webkit-scrollbar-thumb{background-color:#ffffff;border-radius:15px;background-clip:content-box}html ::-webkit-scrollbar-thumb:hover{background-color:#2d2d2d}html ::-webkit-scrollbar-track{background-color:rgba(156,156,156,0.429)}.call-design{display:grid;margin:0px 1.5rem;padding:0px 0.5rem !important;grid-template-columns:100%;grid-template-rows:auto}.call-design button{font-size:13px !important;margin:10px 4px !important}#callendar i{pointer-events:none;border-radius:10px;transition:0.15s all}#callendar i:hover{background-color:rgba(3,169,244,0.6) !important}.cal{display:table}.cal td{border-radius:50px !important;width:calc(100% / 7) !important}.cal th,.cal td,.cal tr{border:none !important}.cal-item{width:100%;height:100%;padding:10px;border:rgba(255,193,7,0.6) 2px solid;border-radius:50px;transition:all 0.3s;pointer-events:none}.cal-item:hover{background-color:rgba(255,193,7,0.6);border-color:rgba(255,255,255,0.6)}.text:hover .cal-item{background-color:rgba(255,193,7,0.6);border-color:rgba(255,255,255,0.6)}.car{height:200px;margin:0px 10% 0px 10% !important}.pad{padding:13.2px 0px !important}.hei ul.list-unstyled{max-height:380px;overflow-y:auto}.hei-small ul.list-unstyled{max-height:230px;overflow-y:auto}.hei-xl ul.list-unstyled{max-height:530px;overflow-y:auto}.max-overflow{height:580px !important;max-height:580px !important;overflow:auto !important}.max-overflow1{height:450px !important;max-height:450px !important;overflow:auto !important}.sat{background-

```

color:#0dd12a}.over-f{max-height:calc(650px);overflow:auto !important}.yen{background-color:#e7c745  
 !important}.text{display:table-cell;width:auto;text-align:center;vertical-align:middle}table td{word-wrap:break-  
 word !important;position:relative;min-width:100px}.cal td{min-width:auto !important}.edit{background-  
 color:#d700bb !important;transition:0.3s all}.edit:hover{background-color:#aa0393  
 !important}.prod{background-  
 color:#0aff95}input[type="checkbox"]{position:relative;width:40px;height:20px;-webkit-appearance:none;-  
 webkit-appearance:none;background:#c6c6c6;outline:none;cursor:pointer;border-radius:20px;box-shadow:inset  
 0 0 5px rgba(0,0,0,0.2);transition:background 300ms  
 linear}input[type="checkbox"]::before{position:absolute;content:"";width:20px;height:20px;top:0px;left:0px;bo  
 rder-radius:10px;background-color:#fff;transform:scale(1.1);box-shadow:0 2px 5px  
 rgba(0,0,0,0.2);transition:left 300ms  
 linear}input[type="checkbox"]:checked{background:#1d4b79}input[type="checkbox"]:checked::before{left:20  
 px;background-color:#fff}.nav-link{width:120px !important;text-align:center}#footer{transition:all  
 0.3s;padding:20px !important;background-color:#03A9F4}#footer:hover{background-color:#1d4b79  
 !important}.frm{background-color:#ff8b07;border-radius:50px;width:80%;margin:20px 10%;padding:20px  
 0px;height:auto;transition:all 0.3s}.frm:hover{color:white !important;background-color:#1ABC9C;box-  
 shadow:9px 9px 0px 3px #212121,19px 19px 0px 0px rgba(33,33,33,0.7),28px 28px 0px 0px  
 rgba(33,33,33,0.4),38px 40px 0px 0px rgba(33,33,33,0.1)}.frm a{color:white}.frm textarea{padding:0px  
 15px;outline:none;border:none}.in option{overflow-y:scroll}#blocks{display:grid;height:auto;row-  
 gap:1rem;grid-template-columns:100%;grid-template-rows:auto;margin:0px 10% 0px 10%  
 !important}@keyframes cardA{0%{transform:translateY(10px)}100%{transform:translateY(-10px)}}.stock-  
 cont{display:grid;grid-template-columns:25% 50% 25% !important;grid-template-rows:auto;font-size:13px  
 !important}@media only screen and (max-width: 425px){.login{width:100% !important}#login{width:100%  
 !important}.font{font-size:12px !important}.font h6{font-size:13px !important}.font h5{font-size:15px  
 !important}.font h4{font-size:19px !important}.call-design{display:grid;padding:0px !important;grid-template-  
 columns:100%;margin:0px !important;grid-template-rows:auto auto}.call-design button{font-size:12px  
 !important;margin:10px 4px !important}.table{-webkit-overflow-scrolling:touch;font-size:13px  
 !important}.stock-cont{display:grid !important;grid-template-columns:50% 50% 100% !important;grid-  
 template-rows:auto auto !important}.stock-cont h3{margin:10px auto !important;padding:5px 13px  
 !important}.st-1{grid-row:1 !important;order:1 !important}.st-2{grid-row:2 !important;grid-column:1/3;order:3  
 !important}.st-2 button{font-size:13px !important}.st-3{grid-row:1 !important;order:2  
 !important}.car{height:200px !important;font-size:13px !important;margin:0px !important}#graph  
 h4{width:100%;text-align:center}#group{width:100%;display:grid !important;grid-template-  
 columns:100%;padding:0px !important}#group div{width:100%}#blocks{display:grid;height:auto;padding:0px  
 !important;grid-template-columns:100%;grid-template-rows:auto auto;font-size:13px !important;row-  
 gap:0.25rem !important;margin:0px !important}#blocks h4{font-size:15px !important}#blocks button{font-  
 size:10px !important;padding:5px 0px !important}.cardsAV{flex-direction:column !important;font-size:12px  
 !important}.cardsAV div{width:calc(100% - 20px) !important}.i1{display:none}.i2{display:block  
 !important;font-size:15px}.nav-link{width:50px !important;text-align:center}.cor-text{position:relative  
 !important;font-size:12px !important;top:0px;right:0px;width:100% !important;height:300px  
 !important;background-color:rgba(0,0,0,0.8);justify-content:start !important;align-items:center}.carousel-  
 item{max-height:calc(100vh - 50px) !important}.carousel-item button{font-size:12px !important}.cal-  
 item{padding:0px !important}.main-opt{position:relative;display:grid;grid-template-columns:25% 25% 25%  
 25%;justify-content:center !important;align-content:center !important;height:auto;background-  
 color:#ff8b07;text-align:center !important;font-size:12px}.main-opt li{display:inline-block;padding:5px  
 0px;transition:all 0.3s}.main-opt2{position:relative;display:grid;grid-template-columns:50% 50%;justify-  
 content:center !important;align-content:center !important;height:auto;background-color:#ff8b07;text-  
 align:center !important;font-size:12px}.main-opt2 li{display:inline-block;padding:5px 0px;transition:all



```

0.3s}.Sitem{background-color:#1d4b79;width:100%;height:100%;animation:alternate 0.3s tr}@keyframes
tr{0%{background-color:#4331e8}25%{background-color:#4331e8}50%{background-
color:#13069e}100%{background-color:#1d4b79}}@media only screen and (max-width: 900px) and (min-
width: 425px){.car{font-size:13px;margin:0px 25px !important}.car h5{font-size:15px}.login{width:60%
!important}#login{width:60% !important}.call-design{display:grid;padding:0px 0.5rem !important;grid-
template-columns:100%;margin:0px 1.5rem !important;row-gap:20px;grid-template-rows:auto}.call-design
button{font-size:13px !important;margin:10px 4px !important}#blocks{display:grid;height:auto;padding:0px
!important;grid-template-columns:100%;grid-template-rows:auto 350px;font-size:13px !important;row-
gap:0.25rem !important;margin:0px 25px 0px 25px !important}#blocks button{font-size:10px
!important;padding:5px 0px !important}}@media only screen and (max-width: 995px) and (min-width:
768px){.cardsAV{width:100% !important;grid-auto-columns:calc(50% - 20px) !important;display:grid
!important;grid-template-columns:calc(50% - 20px) calc(50% - 20px)}.cardsAV div{width:100%
!important}}.cardsAV{display:flex;width:100%;padding-left:20px;flex-direction:row;gap:20px}.cardsAV
div{width:calc(25% - 20px);position:relative;transition:0.3s all ease-in-out}.cardsAV div:hover{border-
color:white !important;box-shadow:rgba(50,50,93,0.25) 0px 13px 27px -5px,rgba(0,0,0,0.3) 0px 8px 16px -
8px;animation:cardA 0.2s linear forwards;background-color:#1ABC9C
!important;color:white}.op{animation:op forwards
0.3s}.cross{top:20px;right:20px;position:absolute;color:white}.act{background-color:#ff8b07;animation:bg
alternate 0.3s}@keyframes op{0%{opacity:0.1}25%{opacity:0.3}50%{opacity:0.6}100%{opacity:1}}*{font-
family:'Montserrat', sans-serif !important}.carousel-item{position:relative;width:100%;max-height:calc(100vh -
80px) !important}.i2{display:none}#btn{padding:10px 40px}@media only screen and (max-width: 768px) and
(min-width: 425px){table{table-layout:auto;width:100%;display:table}.font{font-size:14px !important}.font
h6{font-size:15px !important}.font h5{font-size:18px !important}.font h4{font-size:22px !important}.call-
design{padding-left:20px !important;display:grid;margin:0px !important;grid-template-
columns:calc(100%);row-gap:15px;grid-template-rows:auto}.call-design button{font-size:13px
!important;margin:10px 4px !important}.stock-cont{display:grid !important;grid-template-columns:50% 50%
100% !important;grid-template-rows:auto auto !important}.stock-cont h3{margin:15px auto
!important;padding:5px 13px !important}.st-1{grid-row:1 !important;order:1 !important}.st-2{grid-row:2
!important;grid-column:1/3;order:3 !important}.st-2 button{font-size:13px !important}.st-3{grid-row:1
!important;order:2 !important}.car{font-size:13px;margin:0px 25px !important}.car h5{font-size:15px}.text
button{font-size:13px !important}.cardsAV{width:100% !important;grid-auto-columns:calc(50% - 20px)
!important;display:grid !important;grid-template-columns:calc(50% - 20px) calc(50% - 20px)}.cardsAV
div{width:100% !important}.cardsAV div p{font-size:14px}.cor-text{position:relative
!important;top:0px;right:0px;width:100% !important;height:300px !important;background-
color:rgba(0,0,0,0.8);justify-content:start !important;align-items:center}.carousel-item{max-height:calc(100vh -
50px) !important}.i1{display:none}.i2{display:block;font-size:20px}.nav-link{width:40px !important;text-
align:center}.navbar-nav{display:flex !important;flex-direction:row !important}#btn{padding:5px 15px}table{-
webkit-overflow-scrolling:touch}}table{width:100%;display:table}.navbar{background:#03a9f4
!important}.carousel-control-prev,.carousel-control-next{pointer-events:none}.left{pointer-
events:all;position:absolute;top:calc((100% - 15%) / 2);left:10%;width:40%;height:15%;border-
radius:50px;display:flex;align-items:center;justify-content:center;background-
color:rgba(0,0,0,0.5)}.right{pointer-events:all;position:absolute;top:calc((100% - 15%) /
2);right:10%;width:40%;height:15%;border-radius:50px;display:flex;align-items:center;justify-
content:center;background-color:rgba(0,0,0,0.5)}.list-item{cursor:pointer}.hov{transition:.3s all
!important}.hov:hover{color:#03A9F4 !important;background-color:#ffff !important;scale:1.2}.cor-
text{position:absolute;text-align:center;display:flex;flex-direction:column;justify-content:center;align-
items:center;color:white;padding:0px 20px;top:0px;right:0px;width:40%;height:100%;background-
color:rgba(0,0,0,0.8)}.small{animation:smaller forwards 2s}@keyframes

```

```
smaller{0%{width:100%}25%{width:85%}50%{width:65%}100%{width:40%}}#login{position:fixed;top:50%;left:50%;transform:translate(-50%,-50%);margin-top:-50px;margin-left:-50px;z-index:999;background-color:rgba(0,0,0,0.9);width:40%}.login{position:fixed !important;top:50% !important;left:50% !important;transform:translate(-50%,-50%) !important;margin-top:-50px !important;margin-left:-50px !important;z-index:999 !important;background-color:rgba(0,0,0,0.66) !important;width:40%}.config{top:30vh;z-index:3;left:15%;background-color:rgba(0,0,0,0.9) !important;width:70%}.succ{background-color:#11d411 !important;transition:all 0.15s}.succ:hover{background-color:#0e970e !important}.suc{color:#006605 !important}.wrong{color:#f60909}.f-but{font-size:12px !important}.tab-ana{border-collapse:separate;border-spacing:0px 10px}#log{background-color:transparent;color:white}.log{background-color:transparent;color:white}#reg{background-color:transparent;color:white}.in{outline:none !important;padding:3px 15px !important;width:100% !important}.blocker{z-index:2;background-color:transparent;position:fixed;width:100%;top:0px;height:150vh}.p-25{padding:0px 15%}.c{cursor:pointer !important}.btn-width{width:150px}.max-width{max-width:170px !important}.card-bg{background:linear-gradient(to right bottom, rgba(255,255,255,0.411765), rgba(204,204,204,0.196078))}@keyframes bg{0%{background-color:white}25%{background-color:yellow}50%{background-color:#9bff04}100%{background-color:#03A9F4}}#graph{position:relative;overflow-y:hidden;padding:10px !important}.msj-pos{top:0px;right:5px}#analysis{position:relative;max-height:350px;overflow-y:auto}.aktif{border:1px solid white !important}.aktif:active{border:1px solid white !important}.smaller-day{background-color:#d2d2d2 !important;pointer-events:none !important}.sip-day{background-color:#08a43653 !important;pointer-events:all !important}.small-sip-day{background-color:#2422ac9e !important;pointer-events:all !important}@import"../node_modules/bootstrap/scss/bootstrap.scss";
```

## 9.Kod Açıklamaları

### ----app.routing.module.ts----

Bu sayfadaki kodlar sayfaların ile url uzantılarının ayarlandığı sayfadır.

### ----navbar.component.ts----

Bu sayfadaki kodlar bütün sayfalarda bulunan menü çubuğudur. Bu sayfa bileşenden bileşene değişiklik gösterir burada aktif zaman, otomatik logout ve giriş ekranı kodlarını bulundurur.

### ----app.module.ts----

Bu sayfadaki kodlar uygulamanın temel modülüdür. Burada bileşenler, alt modüller, kullanılacak teknolojiler tanımlanır.

### ----yonetici.component.ts----

Bu sayfadaki kodlar yönetici ekranındaki grafik ve analiz bölümü için yazılmıştır burada tavsiye paneli ve grafik analiz panellerinin kodları vardır. Filtreleme sistemlerinin (tarih, kategori, marka filtreleme) kodları da aynı şekilde kodları bu sayfadır.

-----

#### ----yonet-rapor.component.ts----

Bu sayfada raporlar bölümündeki bilgi bloklarının analiz verilerinin hesaplanması ve yansıtılması ve etkileşimlerin kodları yazılmıştır.

-----

#### ----gecmis.component.ts----

Bu sayfadaki kod işlemi tamamlanan siparişlerin getirilmesini ve işlendiği verilerin sayfaya yansıtılmasını sağlar. İçerik hazır olarak başka bir sayfadan çekilmektedir.

-----

#### ----yonet-hesap.component.ts----

Bu sayfadaki kodlar hesaplar paneli ve aktiflik panelindeki işlemlerini gerçekleştiren TS kodlarını içerir.

-----

#### ----sell-calendar.component.ts----

Bu sayfadaki kodlar işlemci de bulunan hazırlanan siparişler bölümündeki takvim paneline aittir. Burada API'den gelen verilere göre takvim üzerindeki veriler işlenir ve işlenen veriye göre görüntü verilir.

-----

#### ----product.component.ts----

Bu sayfada ürünlerin gösterildiği üzerinde işlem yapılığı yapının panelin TS kodları bulunmaktadır.

-----

#### ---- islemci.component.ts----

Bu sayfada işlemci bölümünün İstekler panelini içermektedir burada siparişler onaylanıp işleme konur yada reddedilir. Bu işlemlerin yapıldığı TS kodlarını içermektedir.

-----

#### ---- ekle.component.ts----

Bu sayfada işlemci bölümünün ürünler ve ayrıntılar panelini içermektedir burada ürün ekleme, kaldırma, düzenleme gibi işlemler yapılır. Aynı zamanda marka ve kategori de eklenebilir. Bu işlemlerin yapıldığı TS kodlarını içermektedir.

-----

#### ----hazirlanan.component.ts----

Bu sayfada sipariş takviminin ve hazırlanan siparişlerin verilerini sağlayan TS kodları bulunmaktadır.

#### ----kayit.component.ts----

Bu sayfada ürün stoğu arttırımı ve ürünler sayfasındaki özelliklerin TS kodları bulunmaktadır.

#### ----depo.component.ts----

Bu sayfada depo sipariş işlemini bitirdikten sonra sipariş işlemini tamamlar ya da bir aksilik çıktıysa iptal eder. Bu işlemlerin TS kodları yazılmıştır.

#### ----magaza.component.ts----

Mağaza ekranının ürünler bölümün TS kodlarını içermektedir sepete ekleme işlemi burada gerçekleşir.

#### ----sepet.component.ts----

Mağaza bölümü burada sepetteki ürünlerin siparişini verir. Burada bu işlemin TS kodları yazılmaktadır.

#### ----siparisler.component.ts----

Bu sayfada mağaza ekranı kendi verdiği siparişleri görür. Burada bu işlemin TS kodları yazılmaktadır.

### 10.test ve hataların bulunması.

Denk gelinen hatalar:

Navbar hatası kullanıcı giriş yap yapıldığında çıkış yapamaması.

http isteklerinin hatalı çalışması.

Tarih dönüşümünün datetime kullanamamak.

Anlık fiyat anlık maliyet anlık stok

ürün kaldırma problemi

api kullanımında hatalı veri gönderdik.

Hata sayfasına yönlendirememek.

Çözüm süreci:

navbar çözümünde yapılanlar

http isteklerinde hatalı olduğunda da göstermesini sağladık

tarihi datetime yerine date olarak revize ettik

anlık maliyet fiyat gibi kolonlar ekledik

ürün kaldırılınca -1 diye bir kategori ayarladık

apiye model olarak veri gönderdik

hata ekranını ls bilgisinin içeriğini temizleyerek yaptık

Test sonuçları:

Hataların çözüm süreci

## **11.proje paketleme ve teslim.**

İçerisine kurulumu anlatan bir dosya ve uygulamanın tamamı bir flash'a konuldu ve dijital kopyaları alındı.