|  |  |
| --- | --- |
|  | **İZMİR BAKIRÇAY ÜNİVERSİTESİ**  **MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ**  **BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ** |

**<Shop Point>**

# ****Sistem Gereksinimleri ve Özellikleri Raporu****

2021-2022 GÜZ

BİL 203 & BİL 209 Dönem Projesi

Sürüm 1.0

*Hazırlayan*

*200601033\_Ali\_Bayram*

210601686\_Kadir\_Kağan\_Yüksel

İçindekiler

İçindekiler 2

Revizyon Geçmişi 2

1. Giriş 3

1.1 Amaç 3

1.2 Ürün Kapsamı 3

1.3 Kurallar, Tanımlar ve Kısaltmalar 3

1.4 Referanslar 3

1.5 Rapora Genel Bakış 3

2. Genel Tanım 4

2.1 Ürüne Bakış 4

2.2 Ürün İşlevleri 4

2.3 Kullanıcı Sınıfları ve Özellikleri 4

2.4 Çalışma Ortamı 4

2.5 Tasarım ve Uygulama Kısıtlamaları 5

2.6 Kullanıcı Belgeleri 5

2.7 Varsayımlar ve Bağımlılıklar 5

2.8 Riskler 5

3. Harici Arayüz Gereksinimleri 6

3.1 Kullanıcı Arayüzleri 6

3.2 Donanım Arayüzleri 6

3.3 Yazılım Arayüzleri 6

3.4 İletişim Arayüzleri 6

4. Sistem Özellikleri 7

4.1 <Sistem Özelliği 1> 7

4.2 <Sistem Özelliği 2> 7

5. Yazılım 8

6. Sistem Tasarımı 9

6.1 Varlık-İlişki Diyagramı 9

6.2 İlişkisel Cebir İfadeleri 9

6.3 Sınıf Diyagramı 9

6.4 *CRUD* Matrisi 9

6.5 View, Trigger ve Stored Procedure Kullanımı 9

Ek A: SQL İfadeleri 10

Ek B: Ekran Görüntüleri 11

Ek C: To Be Determined List (TBDL) 12

Revizyon Geçmişi

|  |  |  |  |
| --- | --- | --- | --- |
| **Adı** | **Tarihi** | **Değişiklik Nedeni** | **Sürüm** |
| Taslak | 20.11.2021 | Sistem gereksinimleri ve özellikleri raporu şablonunun oluşturulması ve açıklanması | 1.0 |
| Faz 1 | 30.11.2021 | Sistemin amacını ve diyagramların oluşturulması ve açıklanması | 1.1 |
| Faz 2 | 18.12.2021 | Sistemin sql kısmında tabloları ve web kısmında sitenin tamamlanarak yüklenmesi | 2.0 |
| Final | 02.02.2022 | Sistemde diyagramların revize edilmesi | 3.0 |

## Giriş

Bu bölümde önerilen sistem için tanımlayıcı bilgilere yer verilir.

Müşterilerin sanal alışveriş yapabileceği, platform sayesinde satıcılara ve ürünlere erişerek alışveriş yapılmasına olanak sağlanır.

### Amaç

Bu raporda gereksinimleri belirtilen ürünü tanımlayın. İlgili faydalar, amaçlar ve hedefler dahil olmak üzere belirtilen yazılımın ve amacının kısa bir tanımını ifade edin.

Geliştiriciler, proje yöneticileri, pazarlama personeli, kullanıcılar, testçiler ve rapor yazarları gibi raporun amaçlandığı hedef okuyucu türlerini tanımlayın

Dünyada popülerleşen alışveriş trendiyle ve artan salgınlar yüzünden insanların kalabalığa karışmak istememesinden dolayı bu projenin ilk adımları atılmıştır. Bu proje ile müşterilerin, satıcılarla fiziksel temasta olmadan online bir alışveriş platformunda birleştirilerek faaliyetlerin sistem üzerinden yürütülmesi amaçlanmıştır. Müşterilerin, satıcıların durumlarını, raftinglerini ve diğer ürünlerin yorumlarını değerlendirerek daha güvenli bir satış alanı oluşturulması hedeflenilmiştir.

Bu proje yeni bir ticaret girişimi kurmak isteyenlere, bu alanda uygulama geliştirmek isteyen yazılımcılara, oluşturulan platformların daha güvenli hale getirilmesi için çalışan testçilere, bu projeyi müşterilerin alışveriş tercihlerine göre müşteriye pazarlayan satış danışmanlarına, projenin tüm detaylarını erişebilen ve onları yönetebilen üst kurula ve bu konuda rapor yazacak kişilere hitap eder.

### Ürün Kapsamı

Özellikle bu rapor, tüm sistemin yalnızca bir bölümünü veya tek bir alt sistemini açıklıyorsa, bu rapora söz konusu olan ürünün kapsamını açıklayın.

Yazılımı kurumsal hedeflerle veya iş stratejileriyle ilişkilendirin.

Ayrı bir vizyon ve kapsam belgesi mevcutsa, onu referans edin.

Bu proje satıcılar ve müşterileri bir araya gelmesini sağlamak için bu platformu kullanmaktadır.

Platforma müşteri ve satıcı kişisel bilgilerle kayıt olur. Kayıt olan satıcı elinde bulunan ürünlerin tedarik edildiği firmaların bilgilerini, ürün kategorisini, ürünün fiyatını , ürünün stok durumunu sisteme ekler.

Müşteri , satıcıları ve ürünleri görüntüleyerek beğendiklerini sepete ekler , eklediği ürünleri ödeme türü seçerek ödeme yapabilir. Ödemesi yapılan ürünlerin sipariş detayı ödeme yapıldıktan sonra ekranda çıkar.

* Ödeme işleminin detayına girilmez.

### Kurallar, Tanımlar ve Kısaltmalar

Özel önemi olan yazı tipleri veya vurgulama gibi bu raporu yazarken izlenen standartları veya tipografik kuralları açıklayın. Örneğin, daha yüksek seviyeli gereksinimler için önceliklerin ayrıntılı gereksinimler tarafından miras alınıp alınmadığını veya her gereksinim ifadesinin kendi önceliğine sahip olup olmayacağını belirtin.

Raporu anlamaya yardımcı olacak ve alana özel terim ve kavramın tanımlarını verin ve rapor içinde kullanılan kısaltmaları alfabetik olarak listeleyin.

### Referanslar

Bu raporun atıfta bulunduğu diğer belgeleri veya web adreslerini listeleyin. Bunlar, kullanıcı arayüzü stil kılavuzlarını, sözleşmeleri, standartları, sistem gereksinimleri spesifikasyonlarını, kullanım senaryosu belgelerini veya bir vizyon ve kapsam belgesini içerebilir. Okuyucunun başlık, yazar, sürüm numarası, tarih ve kaynak veya konum dahil olmak üzere her referansın bir kopyasına erişebilmesi için yeterli bilgiyi sağlayın.

Bu rapor şablonu için, “IEEE Std 830-1998: IEEE Recommended Practice for Software Requirements Specifications” belgesi referans alınmıştır.

Arçelik: https://www.arcelik.com.tr/

Asus: https://www.asus.com/tr/

Bosch: https://www.bosch-home.com.tr/

Dell: https://www.dell.com/tr-tr

Kığılı: https://www.kigili.com/

Koton: https://www.koton.com/tr/

Mavi: https://www.mavi.com/

Monster: https://www.monsternotebook.com.tr/

MSI: https://tr.msi.com/

US.POLO.ASSN: https://tr.uspoloassn.com/

### Rapora Genel Bakış

Bu raporun geri kalanının neler içerdiğini ve nasıl bir bölüm düzeninde olduğunu açıklayın. Raporu okumak için genel bakış bölümlerinden (ikinci bölümden) başlayarak ve her bir okuyucu tipine en uygun bölümlerden geçerek bir sıra önerin.

Arayüz gereksinimlerini front-end yazılımcıları inceleyebilir ve düzenleyebilir. MSSQL kısmını da back-end yazılımcıları inceleyebilir ,düzenleyebilir ve gereken sorguları yapabilir. Sistem özelliklerini, sistem mühendisleri düzenleyebilir.

## Genel Tanım

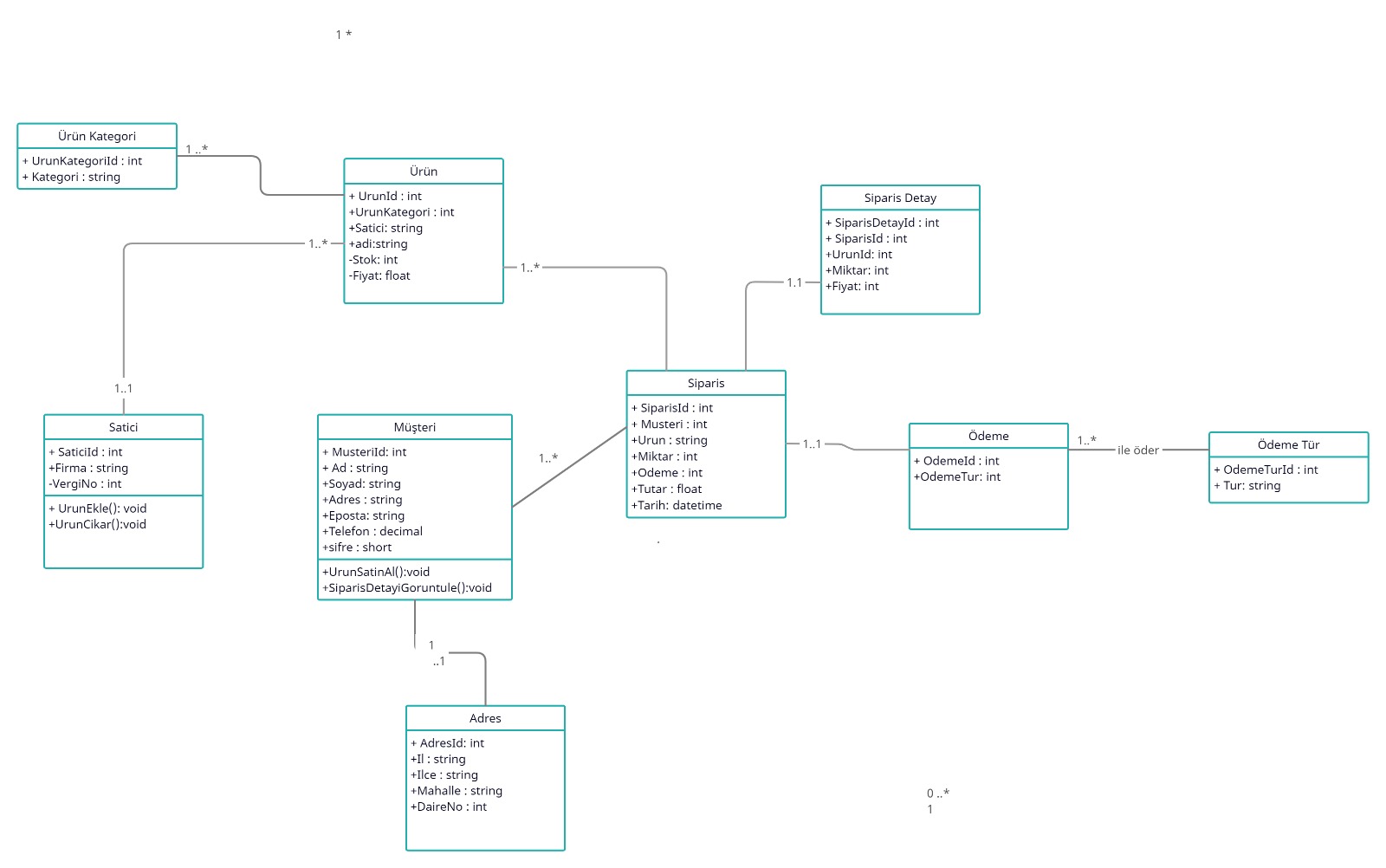
Sistemin içeriğini kısaca tanıtın.

Kullanım senaryolarını ve sistem işlemlerini oluşturun. Bunlar paydaşları belirlemenize yardımcı olacaktır. Paydaşların sistemdeki rolleri nelerdir? Onları listeleyin. Data flow, use case, activity, sequence gibi UML diyagramları çizilebilir.

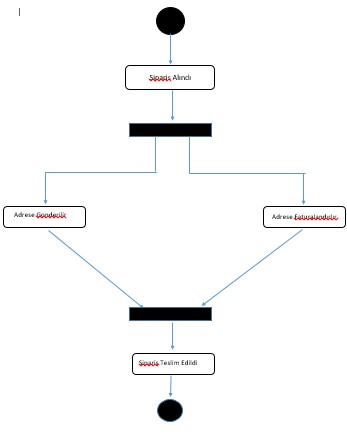
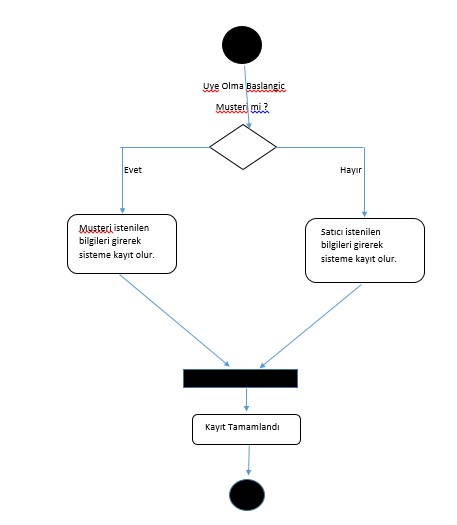
Bu raporda yer alan sistem tasarımı tarafından desteklenen işlevselliğin anlaşılmasına yardımcı olacaksa, bu bölümde projenin arka planını tartışabilirsiniz.

Şekil 1-Use Case Diyagramı

Açıklama: Müşteri ve satıcı sisteme kayıt olabilir. Müşteri sistemdeki ürünleri satın alabilir. Müşteri satın aldığı ürünler için ödeme yapar ve ürünün sipariş detayını görüntüler. Satıcı, müşterinin siparişini teslim eder. Satıcı sistemde sattığı ürünleri çıkarma ve ekleme yapabilir. Müşteri geçmiş siparişleri görüntüleyebilir. Admin satıcı ve müşteri detaylarını görüntüleyebilir.



Şekil 2- UML Diyagramı



Activity Diyagramları

### Ürüne Bakış

Bu raporda belirtilen ürünün bağlamını ve menşeini tanımlayın. Örneğin, bu ürünün bir ürün ailesinin devam üyesi mi, belirli mevcut sistemlerin yedeği mi yoksa yeni, bağımsız bir ürün mü olduğunu belirtin.

Rapor daha büyük bir sistemin bir bileşenini tanımlıyorsa, daha büyük sistemin gereksinimlerini bu yazılımın işlevselliği ile ilişkilendirin ve ikisi arasındaki arayüzleri tanımlayın. Genel sistemin ana bileşenlerini, alt sistem ara bağlantılarını ve harici arabirimleri gösteren basit bir blok diyagram ya da diyagramlar yardımcı olabilir.

Bu proje mevcut alışveriş sitelerinin esinlenerek tasarlanmıştır, Projemiz diğer platformalara göre optimize edilmiştir.

### Ürün İşlevleri

Ürünün gerçekleştirmesi gereken veya kullanıcının gerçekleştirmesine izin vermesi gereken ana işlevleri özetleyin. Ayrıntılar Bölüm 4'te verilecektir, bu nedenle burada yalnızca yüksek düzeyde bir özet (madde işareti listesi gibi) gereklidir. İşlevleri, raporun herhangi bir okuyucusu için anlaşılır kılmak için metinsel ya da grafiksel gösterimleri kullanarak düzenleyebilirsiniz. Bir üst düzey data flow diyagramı veya object class diyagramı gibi, ilgili gereksinimlerin ana gruplarının ve bunların nasıl ilişkili olduğunun bir resmi genellikle etkili olabilir. Söz konusu diyagramlar ürünün tasarımını göstermez, sadece işlevler arasındaki mantıksal ilişkiyi ifade eder.

2.0 kısmında uml ve use-case diyagramları gösterilmiştir.

### Kullanıcı Sınıfları ve Özellikleri

Bu ürünü kullanacağını tahmin ettiğiniz çeşitli kullanıcı sınıflarını tanımlayın. Kullanıcı sınıfları, kullanım sıklığına, kullanılan ürün işlevlerinin alt kümesine, teknik uzmanlığa, güvenlik veya ayrıcalık seviyelerine, eğitim düzeyine veya deneyime göre farklılaştırılabilir. Her kullanıcı sınıfının ilgili özelliklerini tanımlayın. Belirli gereksinimler yalnızca belirli kullanıcı sınıflarıyla ilgili olabilir. Bu ürün için en önemli kullanıcı sınıflarını, tatmin edilmesi daha az önemli olanlardan ayırt edin.

Müşteriler, Satıcılar, Yazılımcılar, Kontrol uzmanları, Admin

Kullanım Sıklığı:

Müşteriler>Satıcılar>Yazılımcılar>Admin>Kontrol uzmanları

Müşteriler ; kayıt olma, Giriş yapma, ürünleri inceleme , Ürün alma, Siparis detayını görüntüleme, Ürünleri Sepete ekleme ve satıcıları görüntüleme fonksiyonlarına sahiptir.

Satıcılar ; Ürün ekleme, Kayıt olma , Giriş yapma fonksiyonlarına sahiptir.

Yazılımcılar ; Veritabanı bağlantılarının denetlenmesini, Web front-end ve back-end tarafında geliştiremeler yapmaları ve gelişen teknoloji ile yazılım daha güncel, dinamik ve kolaylaştırılmasını sağlar.

Admin ; Satıcı ve müşteri bilgilerini görüntüleme ve yazılımcılarla bağlantı kurarak yeni geliştirmeler ister.

Kontrol Uzmanı ; Yazılımın açıklarını bularak projenin daha güvenli konuma getirilmesini sağlar.

### Çalışma Ortamı

Donanım platformu, işletim sistemi ve sürümleri ve uyum içinde bir arada var olması gereken diğer yazılım bileşenleri veya uygulamaları dahil olmak üzere yazılımın çalışacağı ortamı tanımlayın.

Projenin oluşturulurken veri tabanı için MSSQL programı kullanılacaktır. Web sitesi oluşturulurken VS Code (HTML, CSS, JAVASCRİPT, JQUERY) kullanılacaktır. Veri tabanı ve web sitesini ilişkilendirmek için Visual Studio MVC kullanılacaktır. Diyagramlar çizilirken online makerlar kullanılmıştır.

### Tasarım ve Uygulama Kısıtlamaları

Geliştiricilerin kullanabileceği seçenekleri sınırlayacak öğeleri veya sorunları tanımlayın. Bunlar şunları içerebilir: kurumsal veya düzenleyici politikalar; donanım sınırlamaları (zamanlama gereksinimleri, birincil ve ikincil bellek gereksinimleri); diğer uygulamalara arayüzler; kullanılacak belirli teknolojiler, araçlar ve veri tabanları; yedekleme ve kurtarma kısıtları; paralel işlemler; dil gereksinimleri; iletişim protokolleri; Güvenlik Hususları; tasarım sözleşmeleri veya programlama standartları (örneğin, müşterinin kuruluşu teslim edilen yazılımın bakımından sorumlu olacaksa).

### Kullanıcı Belgeleri

Yazılımla birlikte teslim edilecek kullanıcı belgeleri bileşenlerini (kullanım kılavuzları, çevrimiçi yardım ve öğreticiler gibi) listeleyin. Bilinen tüm kullanıcı belgeleri teslim biçimlerini veya standartlarını tanımlayın.

### Varsayımlar ve Bağımlılıklar

Raporda belirtilen gereksinimleri etkileyebilecek varsayılan etkenleri (bilinen gerçeklerin aksine) listeleyin. Bunlar, kullanmayı planladığınız üçüncü şahıs veya ticari bileşenleri, geliştirme veya işletim ortamıyla ilgili sorunları veya kısıtlamaları içerebilir. Bu varsayımların yanlış olması, paylaşılmaması veya değişmesi durumunda proje etkilenebilir. Ayrıca, başka bir projeden yeniden kullanmayı düşündüğünüz yazılım bileşenleri gibi, başka bir yerde (örneğin, vizyon ve kapsam belgesinde veya proje planında) belgelenmemişlerse, projenin dış etkenlere olan bağımlılıklarını da belirleyin.

### Riskler

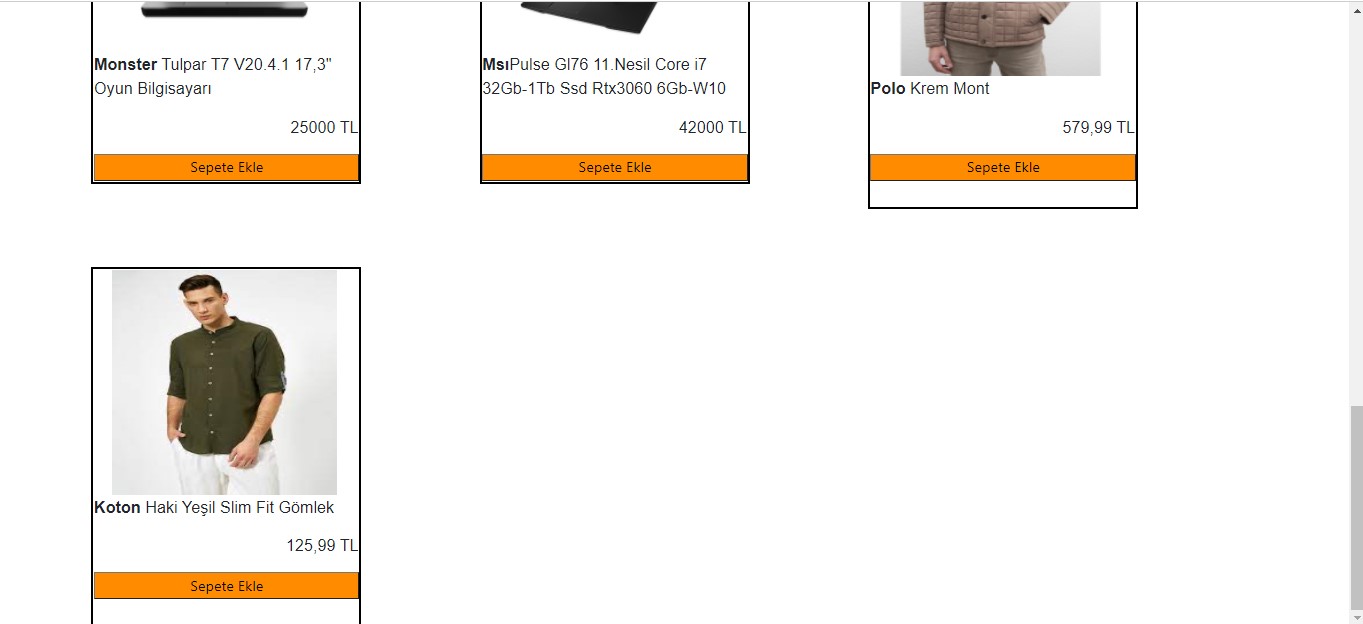
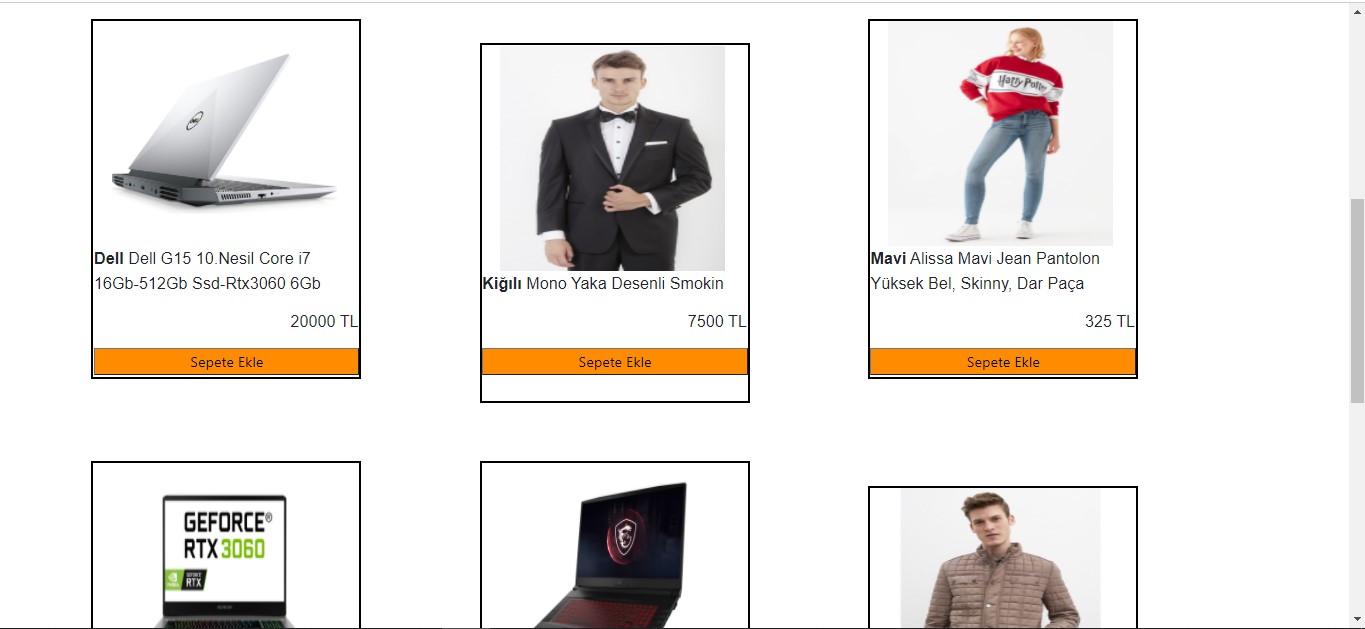
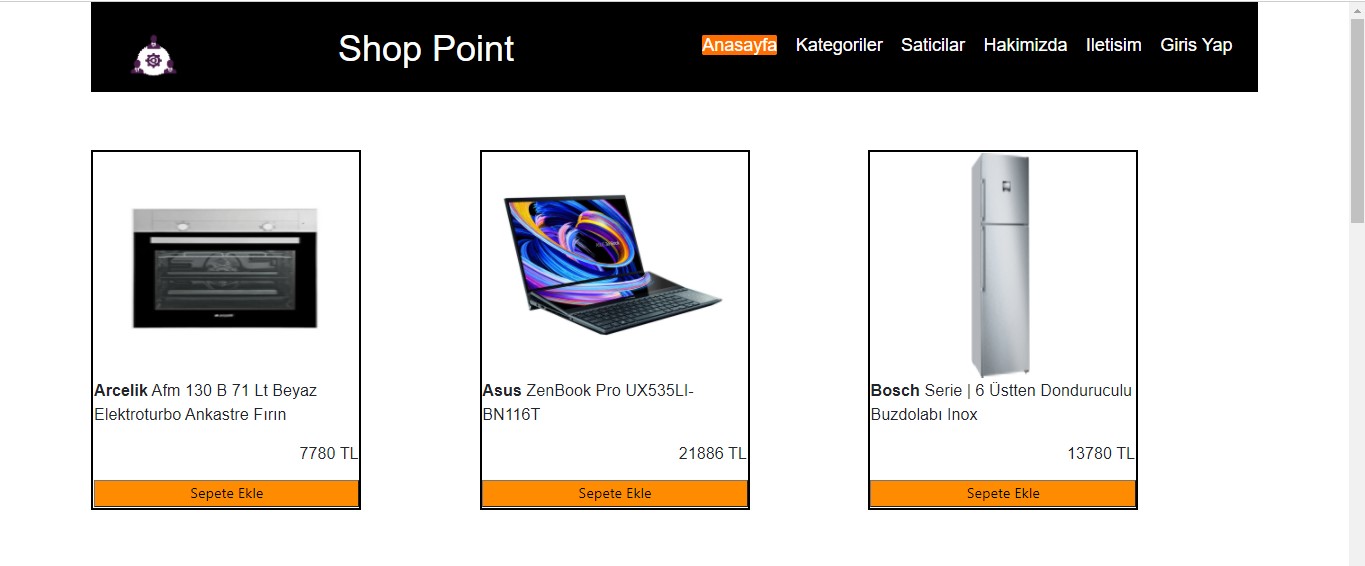
*Sistem tasarımı ve önerilen stratejilerle ilgili tüm riskleri tanımlayın.*

*İşlevsel Olmayan Gereksinimler*

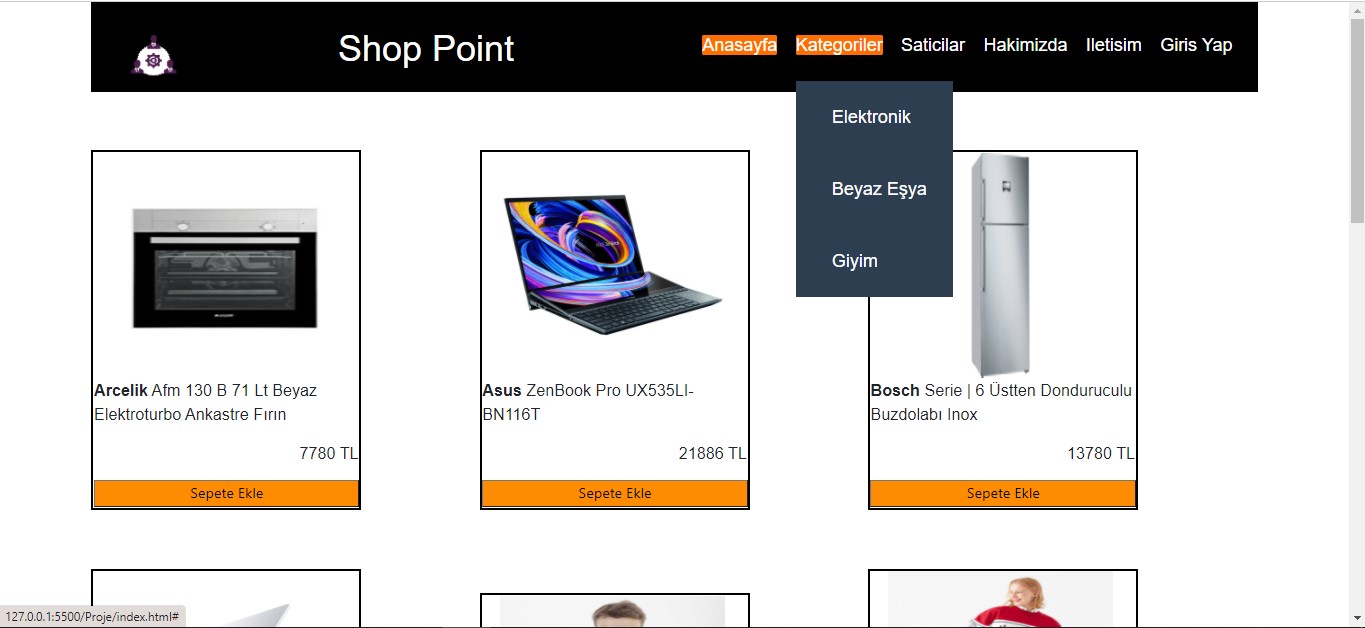
* *Sistemi eş zamanlı olarak 5 milyon kullanıcı kullanacaktır.*
* *MD5 şifrreleme algoritması kullanılarak müşteri ve satıcıya ait veriler korunmalıdır.*
* *Sistem yetkisiz kullanıcı girişlerine karşı %100 güvenli olması.*
* *Sistemin rutin bakımlar nedeniyle kapalı olduğu süre 1 aylık işletim sürecinde 30 dakikayı geçmemesi.*

## Harici Arayüz Gereksinimleri

### Kullanıcı Arayüzleri

Yazılım ürünü ve kullanıcılar arasındaki her bir arabirimin mantıksal özelliklerini tanımlayın. Bu, örnek ekran görüntüleri, herhangi bir GUI standartları veya izlenecek ürün ailesi stil kılavuzları, ekran düzeni kısıtlamaları, her ekranda görünecek standart düğmeler ve işlevler (örn. yardım), klavye kısayolları, hata mesajı görüntüleme standartları ve yakın zamanda. Kullanıcı arabiriminin gerekli olduğu yazılım bileşenlerini tanımlayın. Kullanıcı arayüzü tasarımının detayları ayrı bir kullanıcı arayüzü spesifikasyonundabelgelenmelidir.

Açıklama 1: Anasayfa ürünlerin listelenmesine ve her bir ürünün detaylarının görüntülenmesidir.



Açıklama 2: Anasayfa da navbarda kategorilerin listelenmesidir.



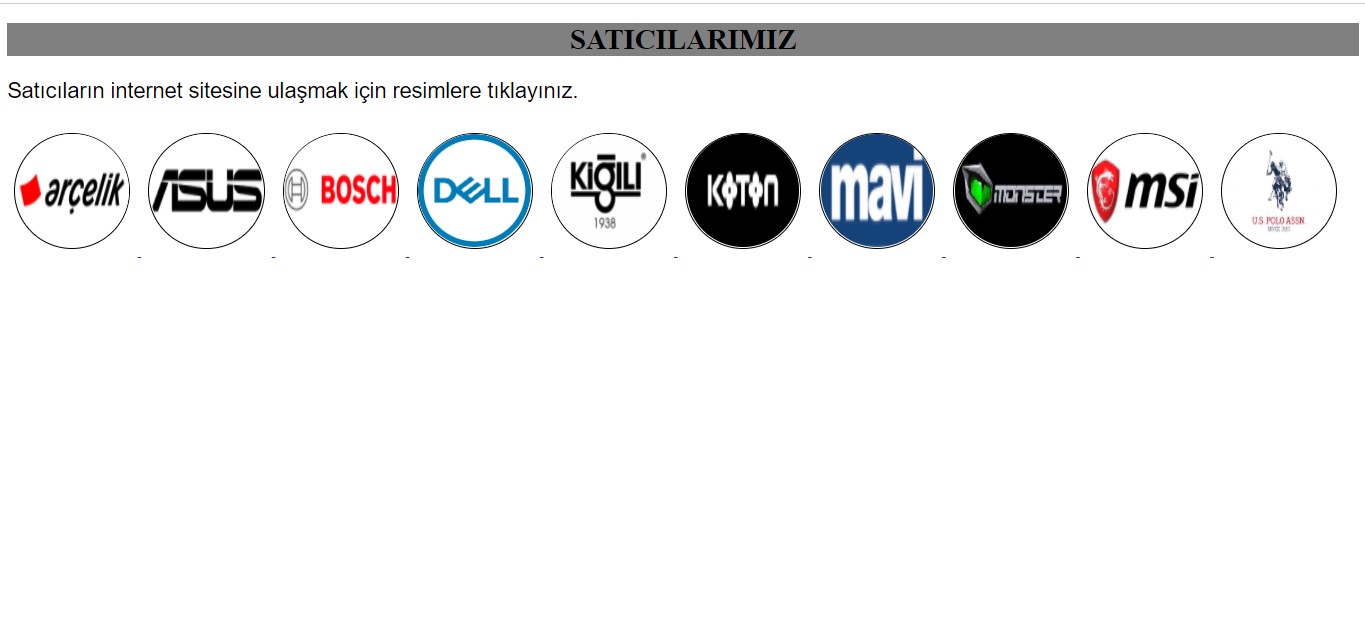
Açıklama 3: Navbardaki giriş yap butonuna tıklandığında ulaştığımız ara yüz ayrıca müşteri ve satıcılar bu alandan üye olma , giriş işlemlerini sağlar.



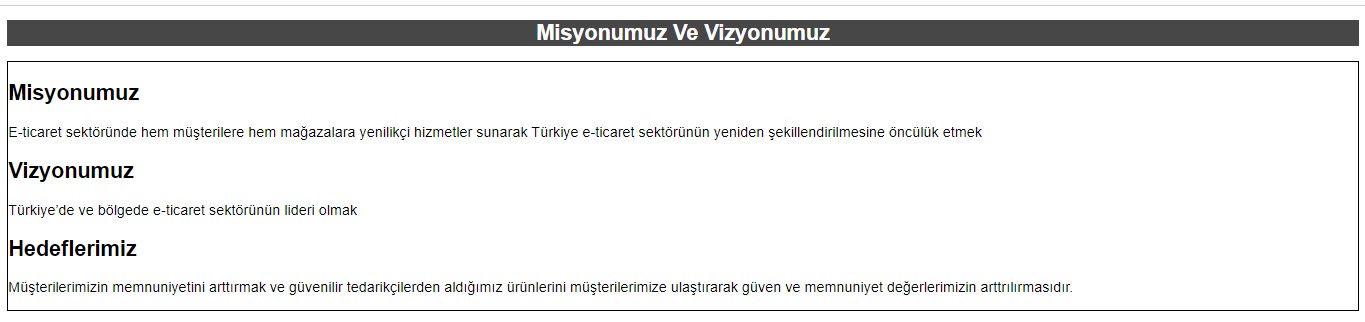
Açıklama 4: Müşteri ve satıcılar giriş arayüzden siteye ulaşım sağlar.



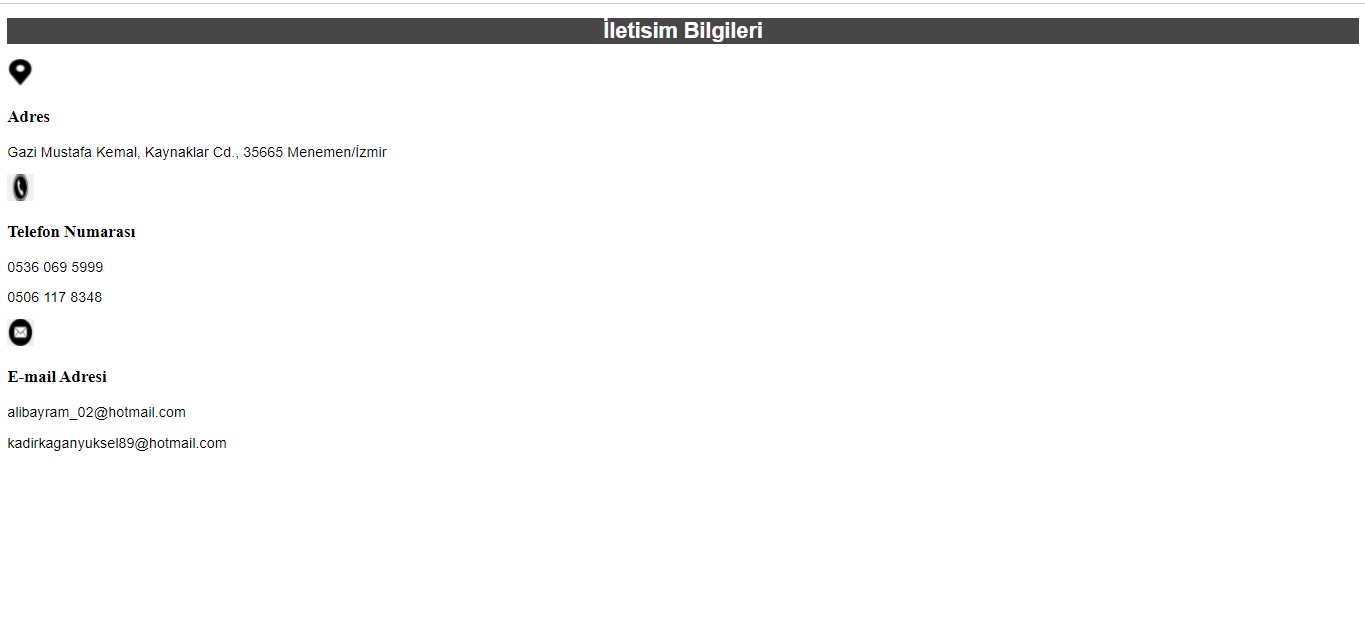
Açıklama 5: Siteye üye olmamış kullanıcıların kayıt olduğu arayüz tasarımıdır.



Açıklama 6: Sistemde yer alan ürünlerin tedarikçileri olan satıcıların listelendiği arayüzdür.



Açıklama 7 : Hakkımızdaki arayüz kısmında misyonumuz , vizyonumuz ve hedeflerimiz yer almaktadır.



Açıklama : Müşteriler ve satıcıların bizimle iletişim kurabileceği ve sorunlarınız olduğunda bize ulaşabileceğiniz arayüzdür.

### Donanım Arayüzleri

Yazılım ürünü ile sistemin donanım bileşenleri arasındaki her arabirimin mantıksal ve fiziksel özelliklerini tanımlayın. Bu, desteklenen cihaz türlerini, verinin yapısını ve yazılım ile donanım arasındaki kontrol etkileşimlerini ve kullanılacak iletişim protokollerini içerebilir.

### Yazılım Arayüzleri

Veritabanları, işletim sistemleri, araçlar, kitaplıklar ve entegre ticari bileşenler dahil olmak üzere bu ürün ile diğer belirli yazılım bileşenleri (ad ve sürüm) arasındaki bağlantıları açıklayın. Sisteme giren ve çıkan veri öğelerini veya mesajları tanımlayın ve her birinin amacını açıklayın. İhtiyaç duyulan hizmetleri ve iletişimin doğasını açıklayın. Ayrıntılı uygulama programlama arabirimi protokollerini açıklayan belgelere bakın. Yazılım bileşenleri arasında paylaşılacak verileri belirleyin. Veri paylaşım mekanizmasının belirli bir şekilde uygulanması gerekiyorsa (örneğin, çok görevli bir işletim sisteminde global bir veri alanının kullanılması), bunu bir uygulama kısıtlaması olarak belirtin.

### İletişim Arayüzleri

E-posta, web tarayıcısı, ağ sunucusu iletişim protokolleri, elektronik formlar vb. dahil olmak üzere bu ürünün gerektirdiği tüm iletişim işlevleriyle ilgili gereksinimleri açıklayın. İlgili mesaj formatını tanımlayın. FTP veya HTTP gibi kullanılacak tüm iletişim standartlarını tanımlayın. Tüm iletişim güvenliği veya şifreleme sorunlarını, veri aktarım hızlarını ve senkronizasyon mekanizmalarını belirtin.

## Sistem Özellikleri

Bu şablon, ürün tarafından sağlanan başlıca hizmetler olan sistem özelliklerine göre ürün için işlevsel gereksinimlerin organize edilmesini gösterir. Bu bölümü, ürününüz için en mantıklı olanı, kullanım durumu, çalışma modu, kullanıcı sınıfı, nesne sınıfı, işlevsel hiyerarşi veya bunların kombinasyonlarına göre düzenlemeyi tercih edebilirsiniz.

### <Sistem Özelliği 1>

Gerçekten “Sistem Özelliği 1” demeyin. Özellik adını birkaç kelimeyle belirtin.

#### Tanım ve Öncelik

Özelliğin kısa bir tanımını yapın ve Yüksek, Orta veya Düşük öncelikli olup olmadığını belirtin. Ayrıca fayda, ceza, maliyet ve risk gibi belirli öncelikli bileşen derecelendirmelerini de dahil edebilirsiniz (her biri en düşük 1 ile en yüksek 9 arasında göreceli bir ölçekte derecelendirilmiştir).

#### Uyaran/Yanıt Dizileri

Bu özellik için tanımlanan davranışı uyaran kullanıcı eylemlerinin ve sistem yanıtlarının sırasını listeleyin. Bunlar, kullanım durumları ile ilişkili diyalog öğelerine karşılık gelir.

#### İşlevsel Gereklilikler

Bu özellikle ilişkili ayrıntılı işlevsel gereksinimleri maddeleyin. Bunlar, kullanıcının özellik tarafından sağlanan hizmetleri yerine getirmesi veya kullanım senaryosunu yürütmesi için mevcut olması gereken yazılım yetenekleridir. Ürünün beklenen hata koşullarına veya geçersiz girdilere nasıl yanıt vermesi gerektiğini dahil edin. Gereksinimler özlü, eksiksiz, açık, doğrulanabilir ve gerekli olmalıdır. Gerekli bilgilerin henüz mevcut olmadığını belirtmek için “TBD”yi (to-be-determined) kullanın. Her gereksinim, bir sıra numarası veya bir tür anlamlı etiketle (REQ-1, REQ-2, …) benzersiz bir şekilde tanımlanmalıdır.

### <Sistem Özelliği 2>

## Yazılım

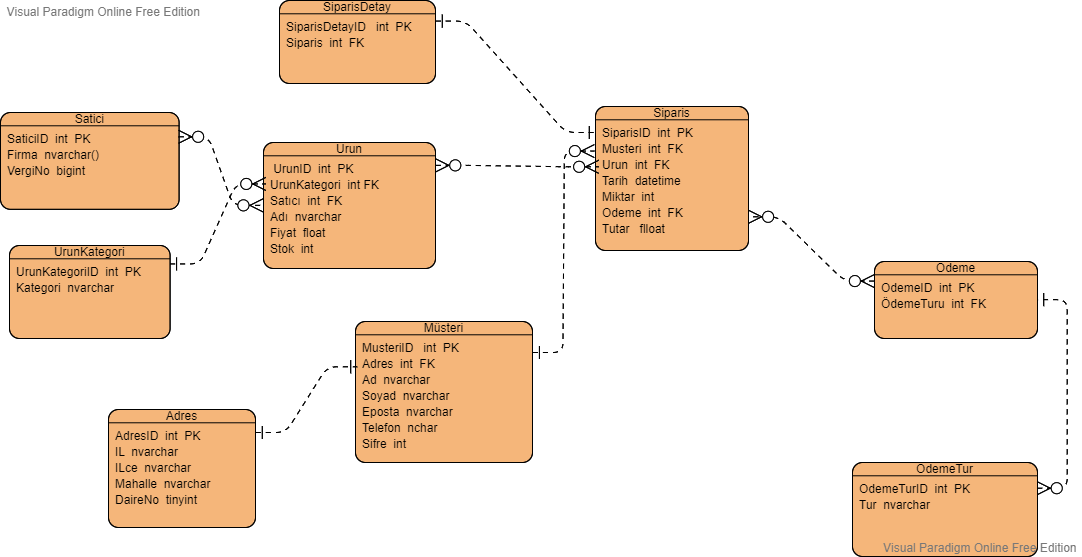
Yazılım mimarisiyle ilgili ayrıntıları gösterin. n-tier mimari kullanıyor musunuz? Projenizde nasıl tasarladınız? Yazılım mimarisi diyagramı çizebilirsiniz (Tipik bir yazılım sisteminin kullanıcıları, dış sistemleri, veri kaynakları ve hizmetleri ile nasıl etkileşime girebileceğini gösterir.). Veritabanı ile ön uç (front-end) arasındaki bağlantıyı nasıl sağladınız? Yazılım mimariniz hakkında ayrıntılı bilgi verin.

Bağlantıyı yapamadık…

## Sistem Tasarımı

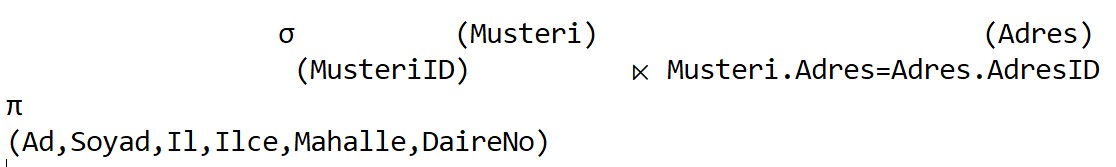
### Varlık-İlişki Diyagramı

*Tüm kuralları (tablolar, özellikler, ilişkiler, kardinaliteler, vb.) uygulayarak Varlık İlişki Diyagramını çizin.*



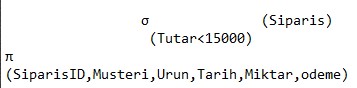
### İlişkisel Cebir İfadeleri

*İlişkisel cebir ifadeleri ve onların SQL cümleleri*



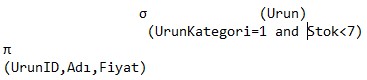
select \* from Musteri

inner join Adres on adres.AdresID=Musteri.Adres;



select SiparisID,Musteri,Urun,Tarih,Miktar,Odeme from Siparis

where Tutar<15000



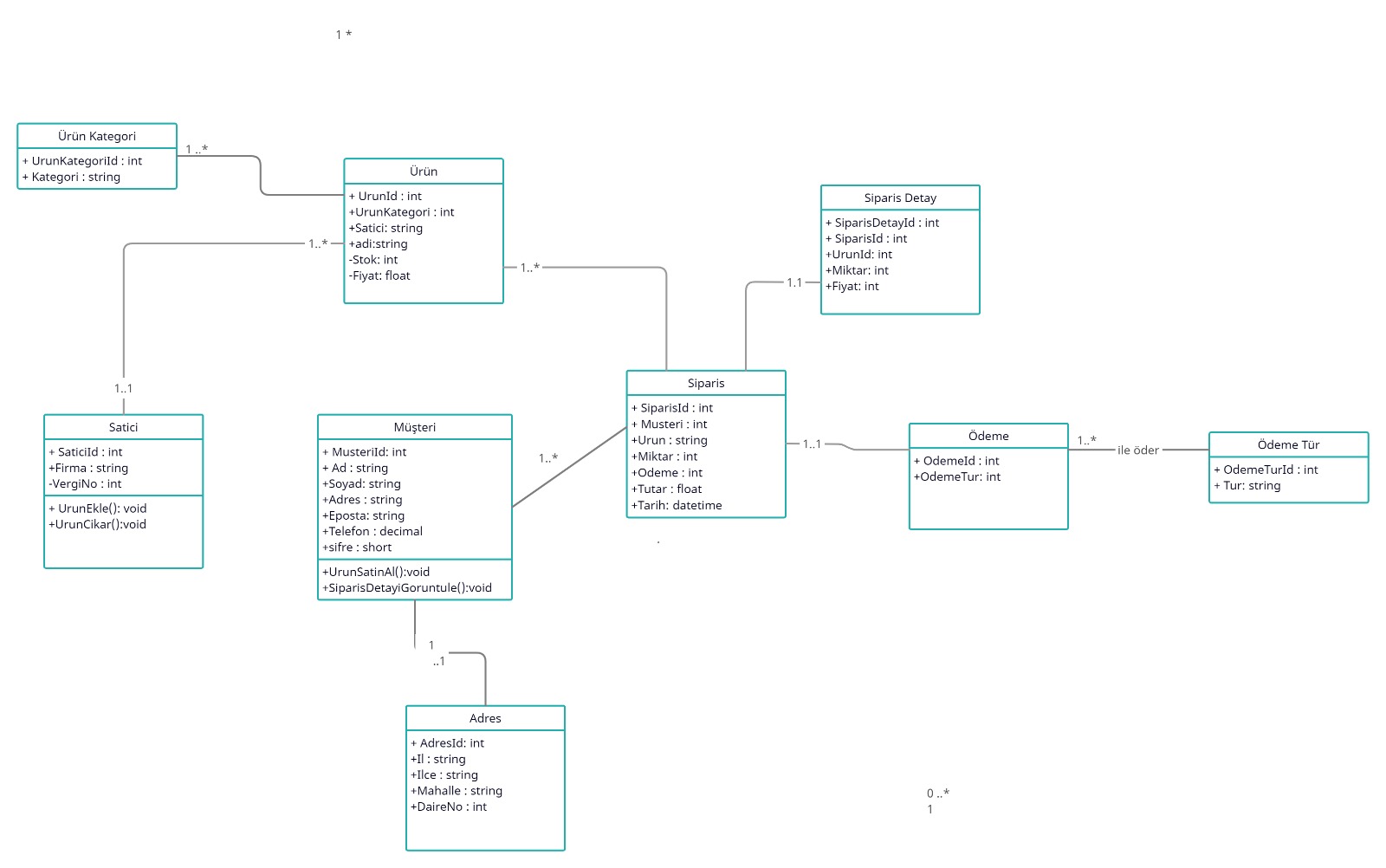
select UrunID,Adı,Fiyat

from Urun

where UrunKategori=1 and Stok<7

### Sınıf Diyagramı

*UML Class diyagramınızı çizin.*

**

### *CRUD* Matrisi

Aşağıdaki tablodaki gibi tüm ilişkiler için verilerin nasıl korunacağını ve verilere nasıl erişileceğini gösteren CRUD Matrisi (Create, Read, Update, Delete) oluşturun. Tüm CRUD işlemlerini tüm tablolara uyguladınız mı?

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tablo-Operasyon** | **CREATE** | **READ** | **UPDATE** | **DELETE** |
| Adres | X | X | X |  |
| Musteri | X | X | X | X |
| Odeme | X |  | X |  |
| OdemeTur | X |  | X |  |
| Satici | X | X | X | X |
| Siparis | X | X | X | X |
| SiparisDetay | X | X |  | X |
| Urun | X | X | X | X |
| UrunKategori | X | X | X |  |
|  |  |  |  |  |
|  |  |  |  |  |

### View, Trigger ve Stored Procedure Kullanımı

*Kullandığınız view, trigger ve stored procedure kullanımlarını açıklayınız.*

view

--Müşteri Hakkında detaylı bilgi veren view

create view MusteriBilgi

as

select MusteriID,ad,Soyad,Eposta,Telefon,Sifre,IL,ILCE,Mahalle,DaireNo from Musteri

inner join Adres

on Musteri.Adres=Adres.AdresID

select \* from MusteriBilgi

--MusteriBilgi view silmek içindir

drop view MusteriBilgi

--Ürünler hakkında detaylı bilgi veren view

CREATE view UrunBilgi

AS

select UrunID,Kategori,Firma,Adı,Fiyat from Urun

inner join UrunKategori

on Urun.UrunKategori=UrunKategori.UrunKategoriID

inner join Satici

on Urun.Satıcı=Satici.SaticiID

select \* from UrunBilgi

--UrunBilgi view silmek içindir

drop view UrunBilgi

-------------------------------

Procedure

--Siparis hakkında procedure

CREATE PROCEDURE SiparisBilgi

as

select SiparisID,Ad,Soyad,Firma,Adı,Miktar,Fiyat,Tutar,Tur

from Siparis

inner join Musteri

on Siparis.Musteri=Musteri.MusteriID

inner join Urun

on Siparis.Urun=Urun.UrunID

inner join Adres

on Musteri.Adres=Adres.AdresID

inner join Satici

on Urun.Satıcı=Satici.SaticiID

inner join Odeme

on Siparis.Odeme=Odeme.OdemeID

inner join OdemeTur

on Odeme.OdemeTuru=OdemeTur.OdemeTurID

--Procedure çalıştırmak içindir

execute SiparisBilgi

--Procedure silmek içindir

drop procedure SiparisBilgi

--SiparisDetayı hakkında procedure

create procedure Detay

as

select SiparisDetayID,Ad,Soyad,Eposta,Telefon,IL,ILCE,Mahalle,DaireNo,Firma,Adı,Miktar,Fiyat,Tur

from SiparisDetay

inner join Siparis

on SiparisDetay.Siparis=Siparis.SiparisID

inner join Musteri

on Siparis.Musteri=Musteri.MusteriID

inner join Adres

on Musteri.Adres=Adres.AdresID

inner join Urun

on Siparis.Urun=Urun.UrunID

inner join Satici

on Urun.Satıcı=Satici.SaticiID

inner join UrunKategori

on Urun.UrunKategori=UrunKategori.UrunKategoriID

inner join Odeme

on Siparis.Odeme=Odeme.OdemeID

inner join OdemeTur

on Odeme.OdemeTuru=OdemeTur.OdemeTurID

--procedure çalıştırmak içindir

execute Detay

--Procedure silmek içindir

drop procedure Detay

-------------------------------

TRIGGER

--Siparis alındığında stok azaltma ve tutar hesaplama trigger kullanımı

create trigger StokAzalt

on Siparis

after insert

as

declare @siparisid int

declare @urunid int

declare @miktar tinyint

declare @urunFiyat float

select @urunid=Urun,@miktar=Miktar from inserted

select @urunFiyat=Fiyat,@siparisid=SiparisID from Urun,Siparis inserted

update Urun set Stok=Stok-@miktar where UrunID=@urunid

update Siparis set Tutar=Miktar\*@urunFiyat where SiparisID=@siparisid

--StokAzaltma triggerını silme

drop trigger StokAzalt

-------------------------------

# Ek A: SQL İfadeleri

Tüm SQL ifadelerine yer verin (komut dosyaları oluşturun ve ekleyin, veritabanındaki temel işlemler, view’ler, karmaşık sorgular, stored procedure’ler, trigger’lar)

Açıklama 1:

Tabloların oluşması ve birincil anahtarların tanımlanması

--Müşteri tablosunun create

create table Musteri(

MusteriID int identity(1,1) primary key,

Adres int,

Ad nvarchar(50),

Soyad nvarchar(50),

Eposta nvarchar(50),

Telefon nchar(30),

Sifre int

);

--Adres tablosunun create

create table Adres(

AdresID int identity(1,1) primary key,

IL nvarchar(25),

ILce nvarchar(25),

Mahalle nvarchar(25),

DaireNo tinyint

)

--Odeme tablosu create

create table Odeme(

OdemeID int identity(1,1) primary key,

OdemeTuru int

)

--OdemeTuru tablosu create

create table OdemeTur(

OdemeTurID int identity(1,1) primary key,

Tur nvarchar(30)

)

--Satıcı tablosu create

create table Satici(

SaticiID int identity(1,1) primary key,

Firma nvarchar(25),

VergiNo bigint

)

--Siparis tablosu create

create table Siparis(

SiparisID int identity(1,1) primary key,

Musteri int,

Satıcı int,

Urun int,

Tarih datetime,

Miktar int,

Odeme int,

Tutar float

)

--SiparisDetay tablosu create

create table SiparisDetay(

SiparisDetayID int identity(1,1) primary key,

Siparis int

)

--Urun tablosu create

create table Urun(

UrunID int identity(1,1) primary key,

UrunKategori int,

Satıcı int,

Adı nvarchar(50),

Fiyat float,

Stok int

)

--UrunKategori tablosu create

create table UrunKategori(

UrunKategoriID int identity(1,1) primary key,

Kategori nvarchar(50)

)

Açıklama 2:

Oluşturulan tablolar için veri girişlerinin yapılması

1-) Adres Veri Girişi:

select \* from Adres

--1

insert into Adres (IL,Ilce,Mahalle,DaireNo)

values ('İzmir','Menemen','Gazi Mustafa Kemal',8);

--2

insert into Adres (IL,Ilce,Mahalle,DaireNo)

values ('Muğla','Yatağan','Cumhuriyet',42);

--3

insert into Adres (IL,Ilce,Mahalle,DaireNo)

values ('Muğla','Fethiye','Akarca',16);

--4

insert into Adres (IL,Ilce,Mahalle,DaireNo)

values ('İzmir','Bornova','Menderes',2);

--5

insert into Adres (IL,Ilce,Mahalle,DaireNo)

values ('İstanbul','Bağcılar','Tefecikler',31);

--6

insert into Adres (IL,Ilce,Mahalle,DaireNo)

values ('Ankara','Beypazarı','Hamidiye',21);

--7

insert into Adres (IL,Ilce,Mahalle,DaireNo)

values ('Antalya','Kalkan','Gündüzler',29);

--8

insert into Adres (IL,Ilce,Mahalle,DaireNo)

values ('Aydın','Çine','Cennet Ayağı',3);

--9

insert into Adres (IL,Ilce,Mahalle,DaireNo)

values ('Antalya','Manavgat','Gazi',23);

--10

insert into Adres (IL,Ilce,Mahalle,DaireNo)

values ('Balıkesir','Edremit','Taşyaka',10);

2-) Müsteri Veri Girişi:

select \* from Musteri

--1

insert into Musteri (Adres,Ad,Soyad,Eposta,Telefon,Sifre)

values(1,'Kadir','Yüksel','Kadiryuksel738@gmail.com',05061178348,1502);

--2

insert into Musteri (Adres,Ad,Soyad,Eposta,Telefon,Sifre)

values(2,'Ali','Bayram','alibayram\_02@hotmail.com',05360695999,7391);

--3

insert into Musteri (Adres,Ad,Soyad,Eposta,Telefon,Sifre)

values(3,'Okan','Öztürkmenoğlu','okan.ozturkmenoglu@bakircay.edu.tr',05558452674,5348);

--4

insert into Musteri (Adres,Ad,Soyad,Eposta,Telefon,Sifre)

values(4,'Esin','Taş','EsinTas3123@gmail.com',05383736403,1545);

--5

insert into Musteri (Adres,Ad,Soyad,Eposta,Telefon,Sifre)

values(5,'Yasemen','Ayber','YasemenAyber8123@gmail.com',05361228452,7192);

--6

insert into Musteri (Adres,Ad,Soyad,Eposta,Telefon,Sifre)

values(6,'Aksel','Göktürk','AkselGoktur\_231@gmail.com',05903123212,5314);

--7

insert into Musteri (Adres,Ad,Soyad,Eposta,Telefon,Sifre)

values(7,'Gülce','Fındık','GulceFındık@gmail.com',05322451006,7913);

--8

insert into Musteri (Adres,Ad,Soyad,Eposta,Telefon,Sifre)

values(8,'Seren','Akdemir','SerenAkdemir@gmail.com',05892313256,9016);

--9

insert into Musteri (Adres,Ad,Soyad,Eposta,Telefon,Sifre)

values(9,'Sultan','Sarızeybek','SultanSarızeybek7894@gmail.com',05328919201,3719);

3-) OdemeTur Veri Girişi:

select \* from OdemeTur

--Kredi kartı

insert into OdemeTur (Tur)

values ('Kredi Kartı');

--Nakit

insert into OdemeTur (Tur)

values ('Nakit');

4-) Satıcı Veri Girişi:

select \* from Satici

--1

insert into Satici (Firma,VergiNO)

values ('Asus',1310779492);

--2

insert into Satici (Firma,VergiNO)

values ('Monster',9158491208);

--3

insert into Satici (Firma,VergiNO)

values ('Koton',8319310213);

--4

insert into Satici (Firma,VergiNO)

values ('Polo',7123719051);

--5

insert into Satici (Firma,VergiNO)

values ('Bosch',8751912362);

--6

insert into Satici (Firma,VergiNO)

values ('Arçelik',2486410126);

--7

insert into Satici (Firma,VergiNO)

values ('Kiğılı',5712852061);

--8

insert into Satici (Firma,VergiNO)

values ('MSI',3897128453);

--9

insert into Satici (Firma,VergiNO)

values ('Dell',9127386172);

--10

insert into Satici (Firma,VergiNO)

values ('Mavi',5189361046);

5-) Siparis Veri Girişi:

--Urun 1

insert into Siparis (Musteri,Urun,Tarih,Miktar,Odeme)

values (1,4,GETDATE(),3,2);

--Urun 2

insert into Siparis (Musteri,Urun,Tarih,Miktar,Odeme)

values (2,2,GETDATE(),1,2);

--Urun 3

insert into Siparis (Musteri,Urun,Tarih,Miktar,Odeme)

values (3,3,GETDATE(),2,1);

--Urun 4

insert into Siparis (Musteri,Urun,Tarih,Miktar,Odeme)

values (4,5,GETDATE(),2,1);

--Urun 5

insert into Siparis (Musteri,Urun,Tarih,Miktar,Odeme)

values (5,7,GETDATE(),1,1);

--Urun 6

insert into Siparis (Musteri,Urun,Tarih,Miktar,Odeme)

values (6,8,GETDATE(),4,1);

--Urun 7

insert into Siparis (Musteri,Urun,Tarih,Miktar,Odeme)

values (7,3,GETDATE(),2,2);

--Urun 8

insert into Siparis (Musteri,Urun,Tarih,Miktar,Odeme)

values (8,5,GETDATE(),1,2);

--Urun 9

insert into Siparis (Musteri,Urun,Tarih,Miktar,Odeme)

values (9,6,GETDATE(),2,1);

--Urun 10

insert into Siparis (Musteri,Urun,Tarih,Miktar,Odeme)

values (9,9,GETDATE(),3,1);

--Urun 11

insert into Siparis (Musteri,Urun,Tarih,Miktar,Odeme)

values (3,10,GETDATE(),3,1);

--Urun 12

insert into Siparis (Musteri,Urun,Tarih,Miktar,Odeme)

values (4,1,GETDATE(),5,1);

select \* from Siparis

6-) SiparisDetay Veri Girişi:

select \* from SiparisDetay

--Siparis 1

insert into SiparisDetay (Siparis)

values(1);

--Siparis 2

insert into SiparisDetay (Siparis)

values(2);

--Siparis 3

insert into SiparisDetay (Siparis)

values(3);

--Siparis 4

insert into SiparisDetay (Siparis)

values(4);

--Siparis 5

insert into SiparisDetay (Siparis)

values(5);

--Siparis 6

insert into SiparisDetay (Siparis)

values(6);

--Siparis 7

insert into SiparisDetay (Siparis)

values(7);

--Siparis 8

insert into SiparisDetay (Siparis)

values(8);

--Siparis 9

insert into SiparisDetay (Siparis)

values(9);

**7- ) UrunKategori Veri Girişi:**

select \* from UrunKategori

--1

insert into UrunKategori (Kategori)

values ('Elektronik');

--2

insert into UrunKategori (Kategori)

values ('Beyaz Eşya');

--3

insert into UrunKategori (Kategori)

values ('Giyim');

**8- ) Ürün Veri Girişi:**

--1

insert into Urun (UrunKategori,Satıcı,Adı,Fiyat,Stok)

values (1,1,'Zenbook Pro',15000,10);

--2

insert into Urun (UrunKategori,Satıcı,Adı,Fiyat,Stok)

values (3,3,'Gömlek',125.99,50);

--3

insert into Urun (UrunKategori,Satıcı,Adı,Fiyat,Stok)

values (2,5,'Buzdolabı',13780,10);

--4

insert into Urun (UrunKategori,Satıcı,Adı,Fiyat,Stok)

values (1,2,'Tulpar',25000,7);

--5

insert into Urun (UrunKategori,Satıcı,Adı,Fiyat,Stok)

values (3,4,'Mont',579.99,15);

--6

insert into Urun (UrunKategori,Satıcı,Adı,Fiyat,Stok)

values (2,6,'Fırın',7780,5);

--7

insert into Urun (UrunKategori,Satıcı,Adı,Fiyat,Stok)

values (1,9,'Gaming G15 5511',20000,8);

--8

insert into Urun (UrunKategori,Satıcı,Adı,Fiyat,Stok)

values (3,10,'Pantalon',325,17);

--9

insert into Urun (UrunKategori,Satıcı,Adı,Fiyat,Stok)

values (1,8,'Pulse GL76',42000,5);

--10

insert into Urun (UrunKategori,Satıcı,Adı,Fiyat,Stok)

values (3,7,'Smokin',7500,17);

select \* from urun

Açıklama 3:

View,trigger ve procedure 6.5 kısmında eklemiş bulunmaktayızdır.

Açıklama 4:

Karışık sorgu kullanımı:

Örnek-1)

--Tutarı 2000 tl altından ve odemesi kredi olan musteriyi sorgulama

select SiparisID,MusteriID,ad,Soyad,Eposta,Telefon,Sifre,Tur from Siparis

inner join Musteri

on Siparis.Musteri=Musteri.MusteriID

inner join Odeme

on Siparis.Odeme=Odeme.OdemeID

inner join OdemeTur

on Odeme.OdemeTuru=OdemeTur.OdemeTurID

where SiparisID=((select SiparisID from Siparis

where tutar<=2000 and Miktar=(select min(Odeme) from Siparis

where tutar<600)))

örnek-2)

--Stok 10 ve 15 arası olan ürününde fiyatı 5000 tl den büyük olması gerekiyor ayrıca bu ürünü hangi müşterininaldığını gösteren sorgu

select SiparisID,MusteriID,ad,Soyad,Adı,Fiyat,stok

from Siparis

inner join Musteri

on Siparis.Musteri=Musteri.MusteriID

inner join Urun

on siparis.Urun=Urun.UrunID

where Urun=(select UrunID

from Urun

where UrunID=(select UrunID from urun

where (Stok>=10 and stok<15) and Fiyat>5000))

Açıklama 5:

Musteri ve adres arasındaki bağlantı:

select \* from Musteri

inner join Adres

on adres.AdresID=Musteri.Adres;

Açıklama 6:

Urun, Satıcı ve UrunKategori arasındaki bağlantı:

select UrunID,Adı,Fiyat,stok,Firma,VergiNO,Kategori from urun

inner join Satici on Satici.SaticiID=Urun.Satıcı

inner join UrunKategori on UrunKategori.UrunKategoriID=urun.UrunKategori;

Açıklama 7:

MusteriID=1 olan müşterinin ad update

update Musteri set Ad='Kağan'

where sifre=1502

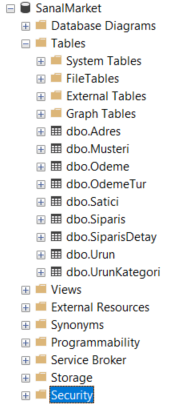
# Ek B: Ekran Görüntüleri

*Ana işlemler için kısa açıklamalı arayüzlerinizin ekran görüntüleri*

*Ek sql kısmındaki açıklamalara göre sıralanacaktır.*

Açıklama 1:

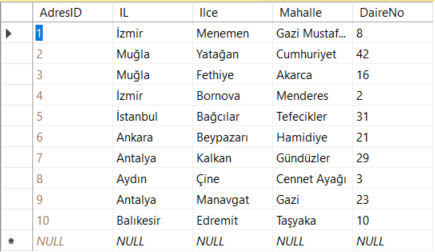
Tabloların oluşturulması



Açıklama 2:

Veri girişlerinin ekran görüntüleri

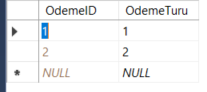
Adres;



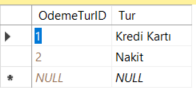
Musteri;



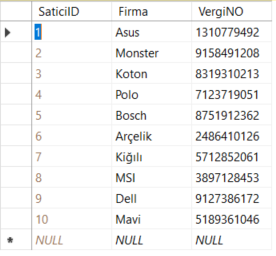
Odeme;



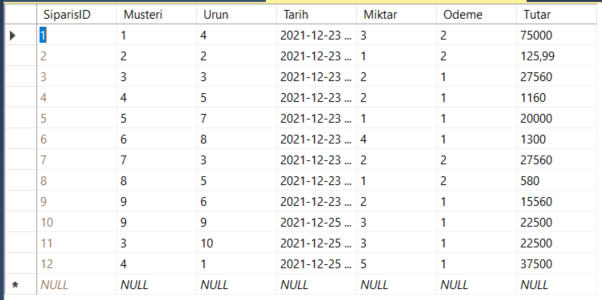
OdemeTur;



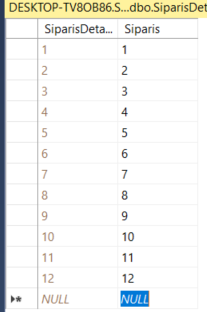
Satıcı;



Siparis;



SiparisDetay;



Urun;



UrunKategorisi;



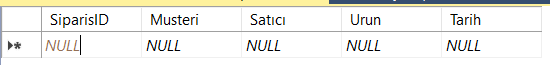
Açıklama 3:

Trigger kullanımı;

Trigger mantığımız siparis alındıktan sonra (insert) hangi urunden kaç tane alınmıs stoktan düşecek ve aldığı üründen miktarına göre tutarını gösterecej o yüzden ilk adım urun ve siparisin ilk halleridir.

6.5 kısmında StokAzaltma trigger kullanımı:

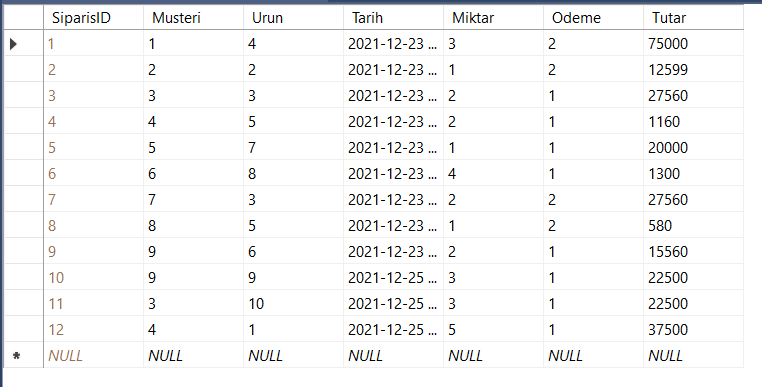
Siparis tablosu (insert ve triggerdan önce):



Urun tablosu (Siparis tablosuna insert ve trigger çalışmadan önce):



Siparis tablosuna (insert) olduktan sonra:

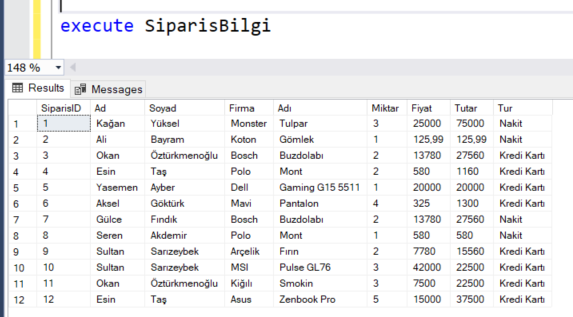


Urun tablosu (siparis tablosuna insert olduktan ve trigger çalıştıktan sonra):

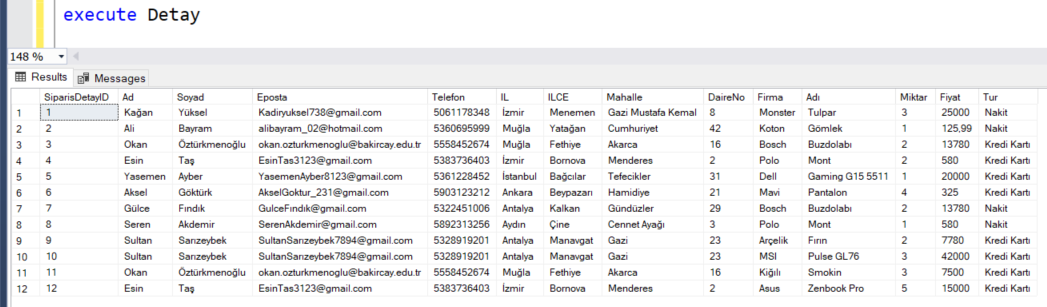


Procedure kullanımı :

her seferinde sorgular yaparak ulaşmak istediğimmz bilgilerin yerine procedureler oluşturarak onları sadece çağırmamız yeterli olucaktır.

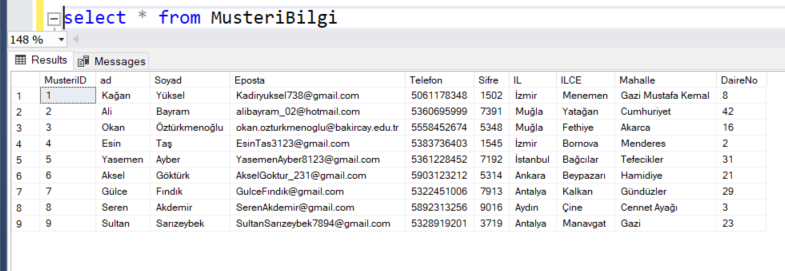
6.5 kısmında kullandığımız SiparisBilgi procedure:

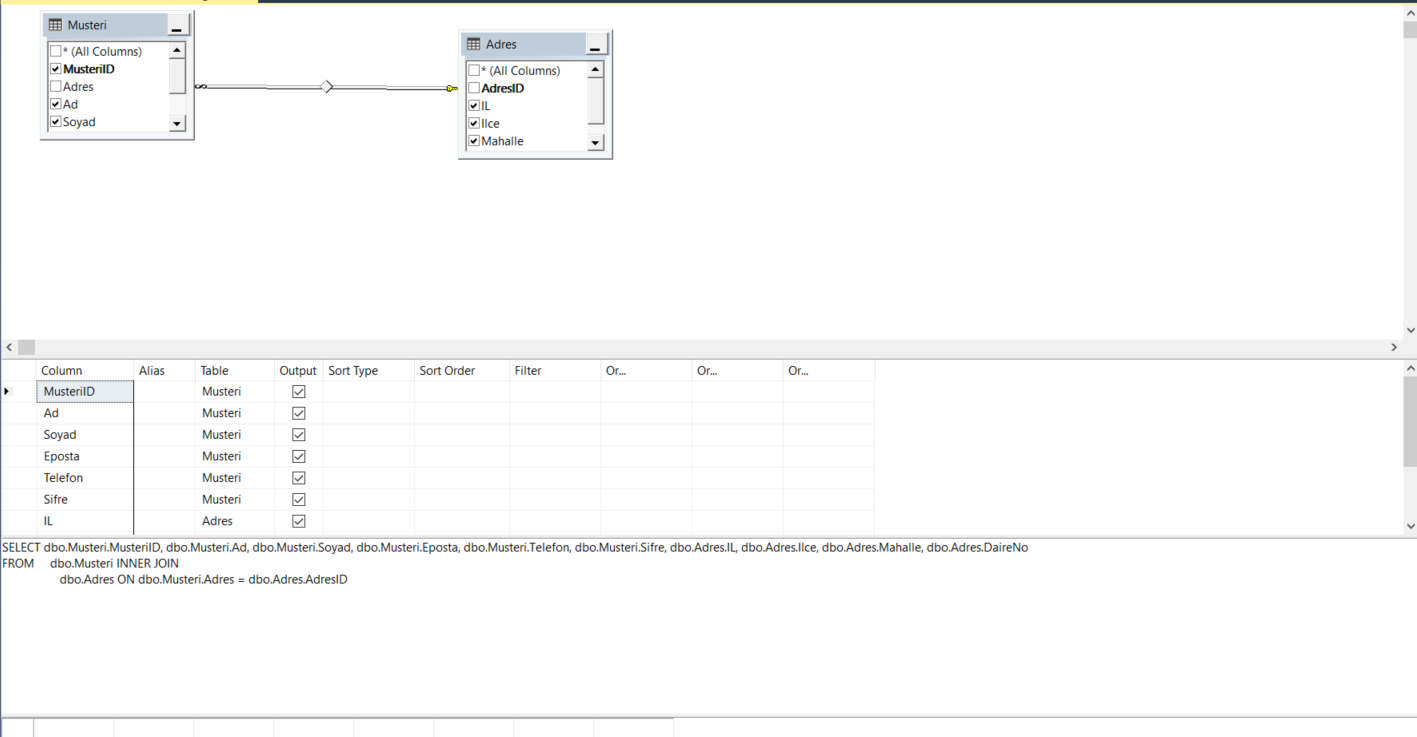
6.5 kısmında kullandığımız Detay procedure:



View Kullanımı: View, kısaca sorgumuzu kayıt altına almaktır. SQL ilişkisel bir veritabanı olduğundan , mimari kurulurken birçok tablo birbiriyle ilişkilendirilir. Bu ilişkilendirme sonucunda istediğimiz veriler farklı tablolar üzerinde olduğunda JOIN kullanmak durumunda kalıyoruz. Bu join yapısı ile iki ya da daha fazla tabloyu birbiriyle bağlayarak tüm sonuçları getirmesini isteyebiliriz.

6.5 kısmında MusteriBilgi View kullanımı:

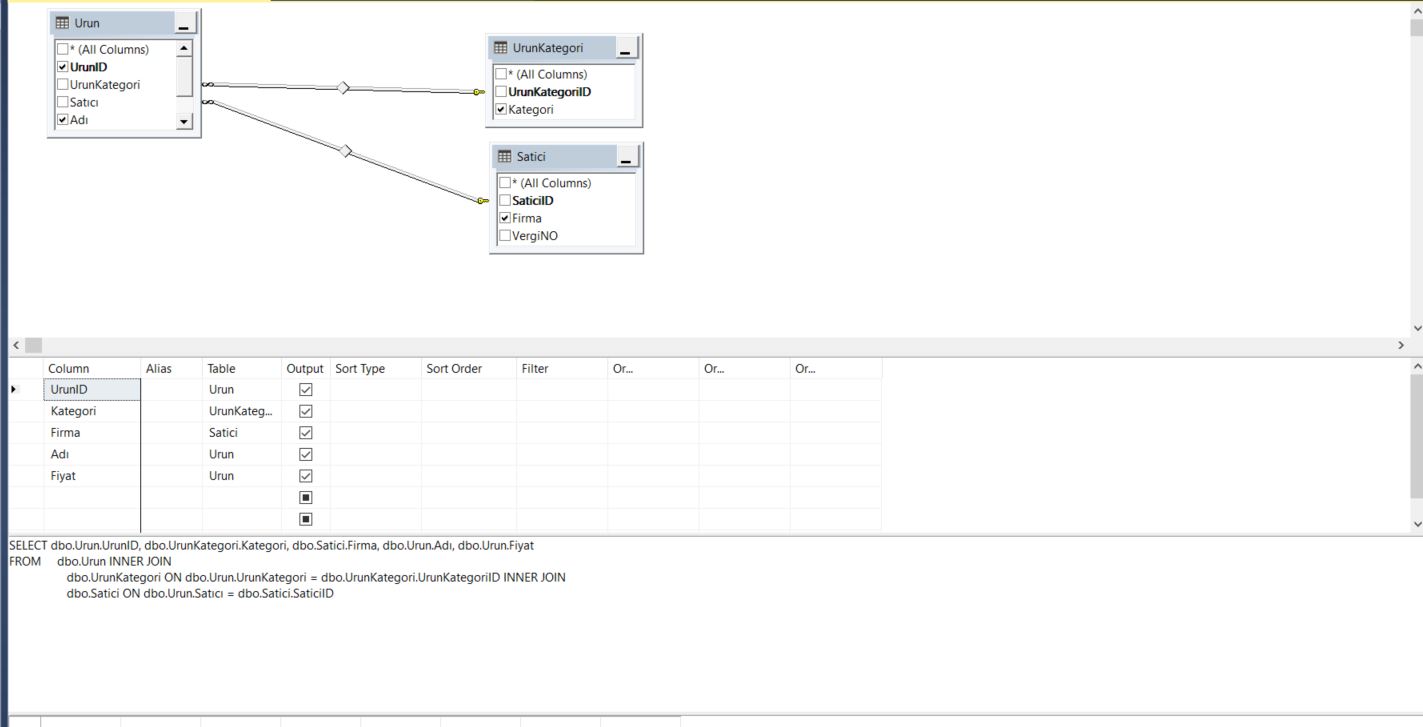




6.5 kısmında UrunBilgi View kullanımı:

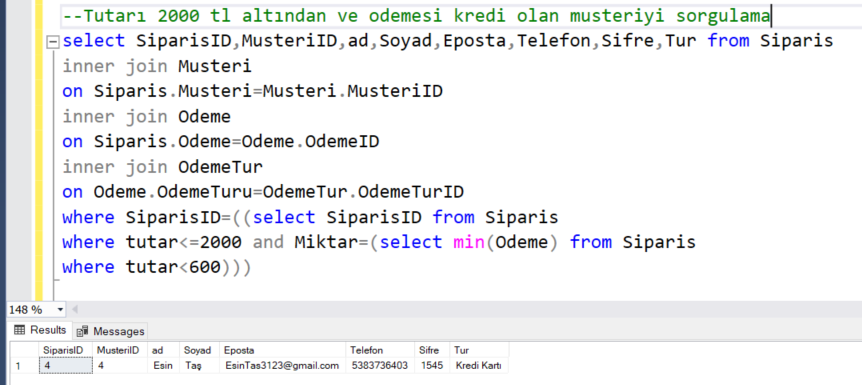
Stok girilmemiştir gizli kalınması istendiği için



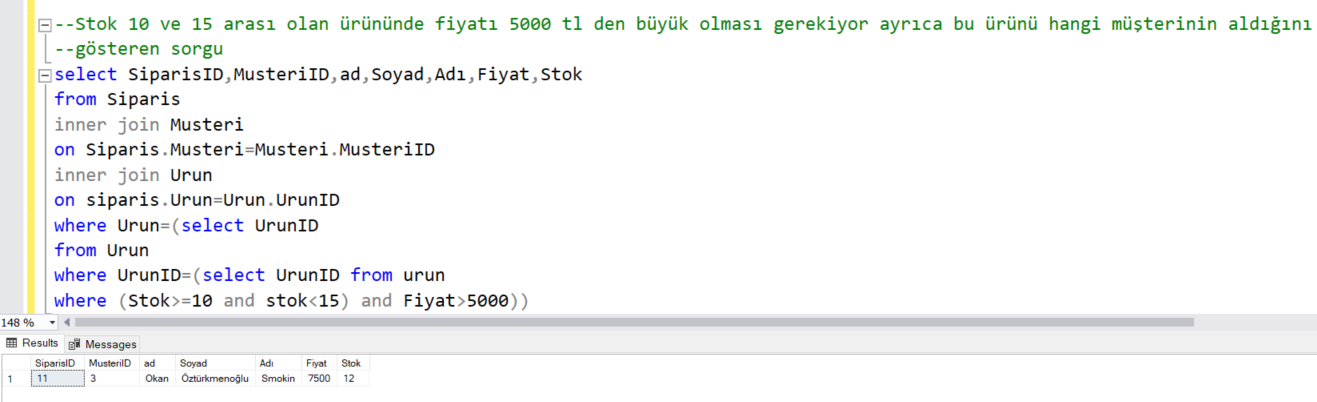


Açıklama 4:

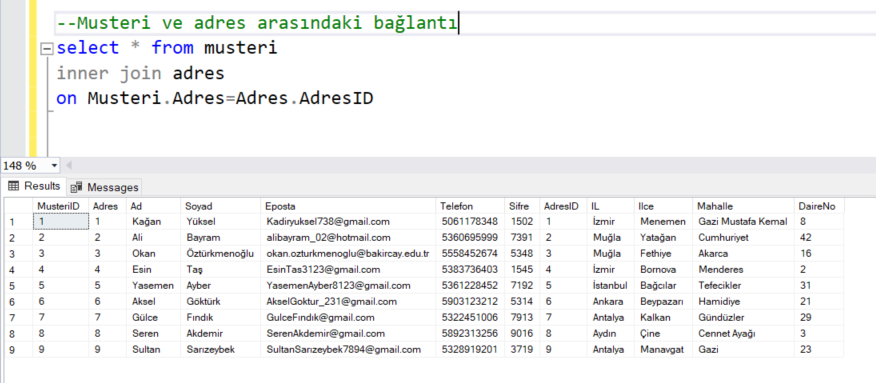
Örnek 1-)



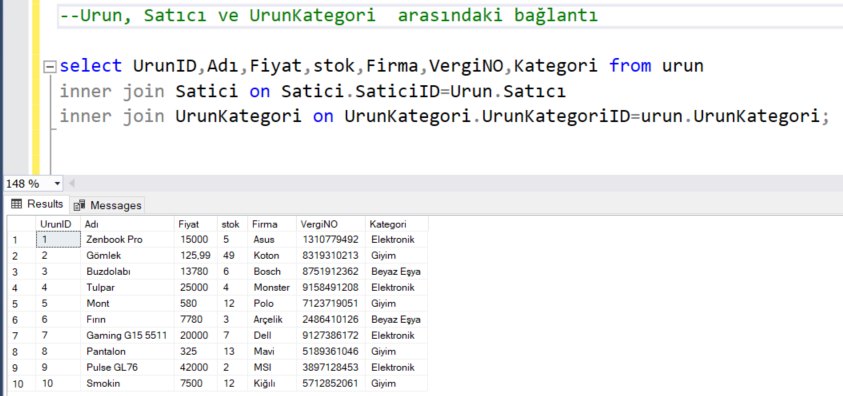
Örnek 2-)



Açıklama 5:

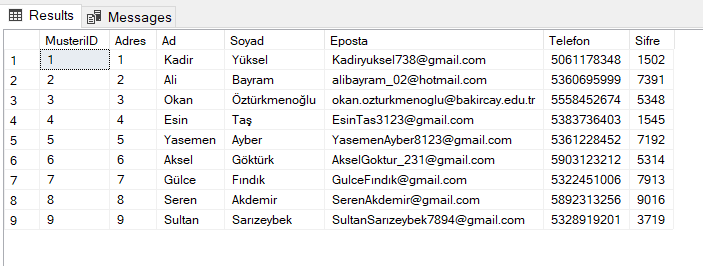


Açıklama 6:

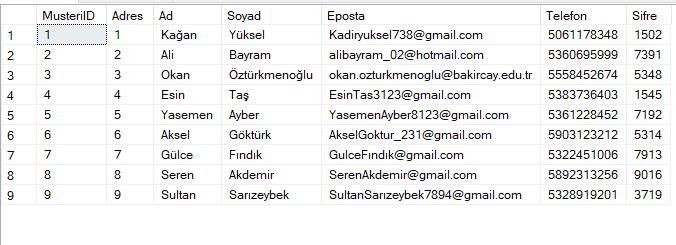
:

Açıklama 7:

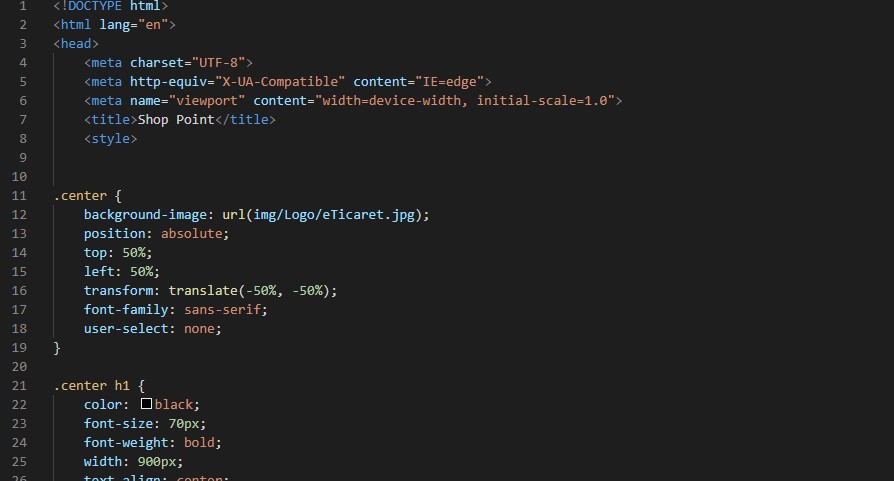
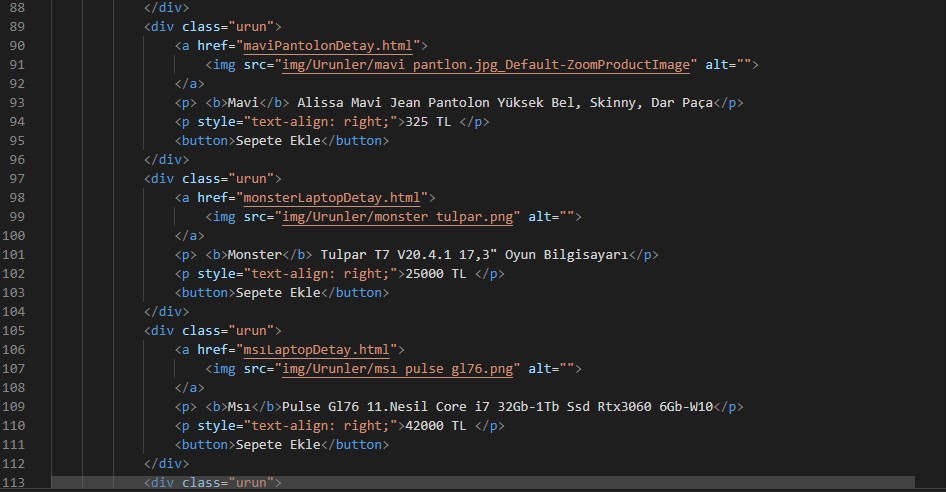
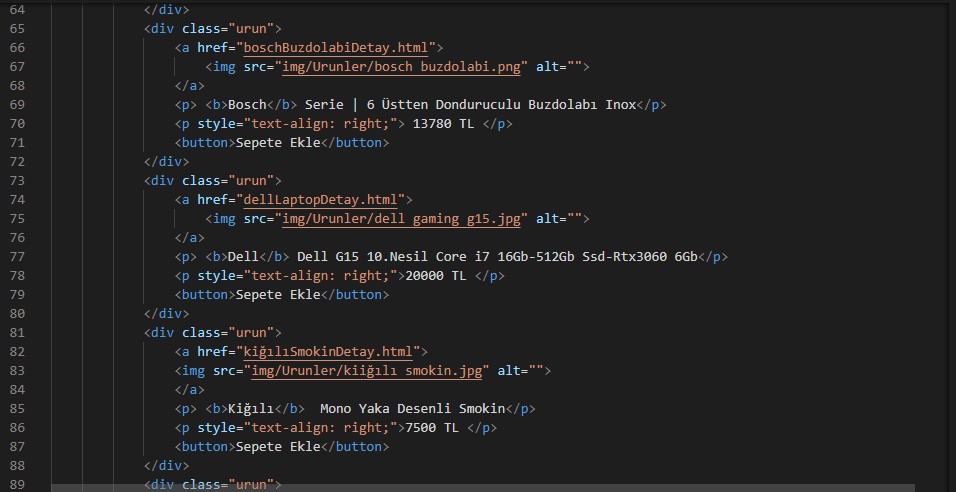
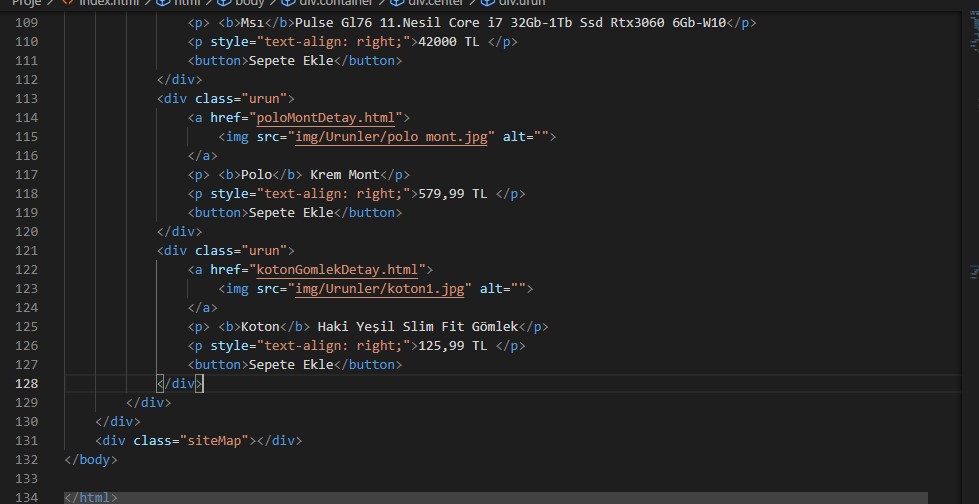
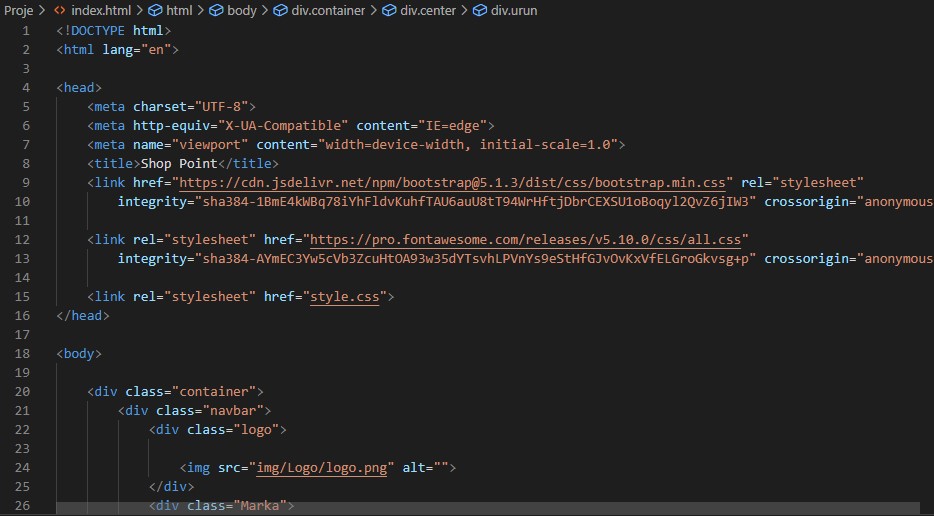
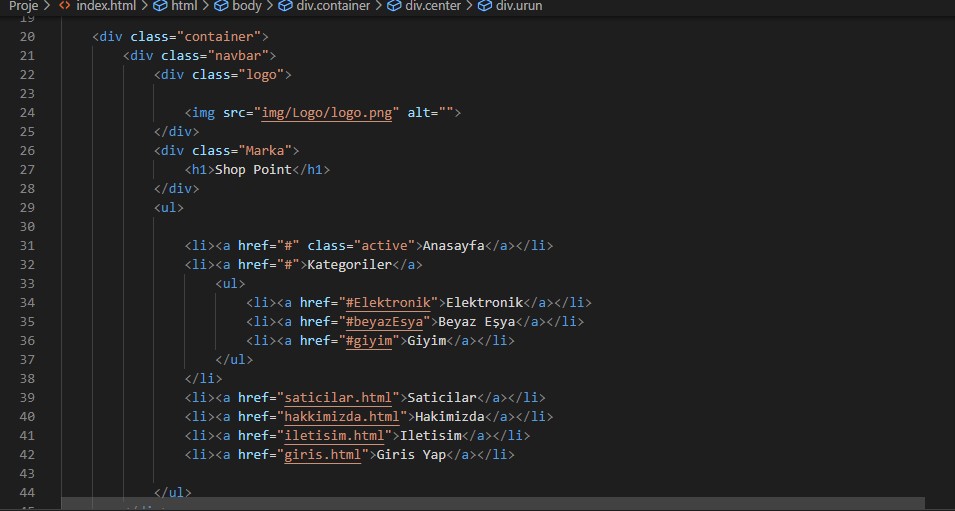
Musteri tablosundaki MusteriID=1 update etmeden önce;

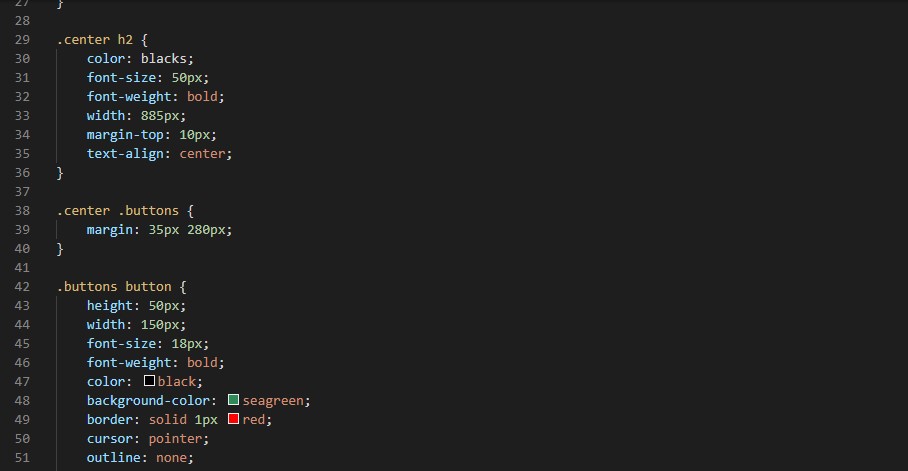


Musteri tablosundaki MusteriID=1 update ettikten sonra;

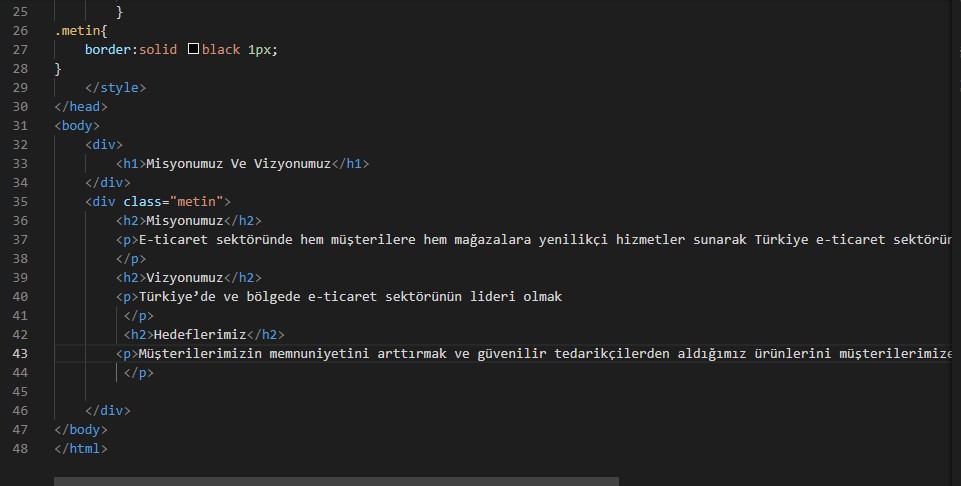
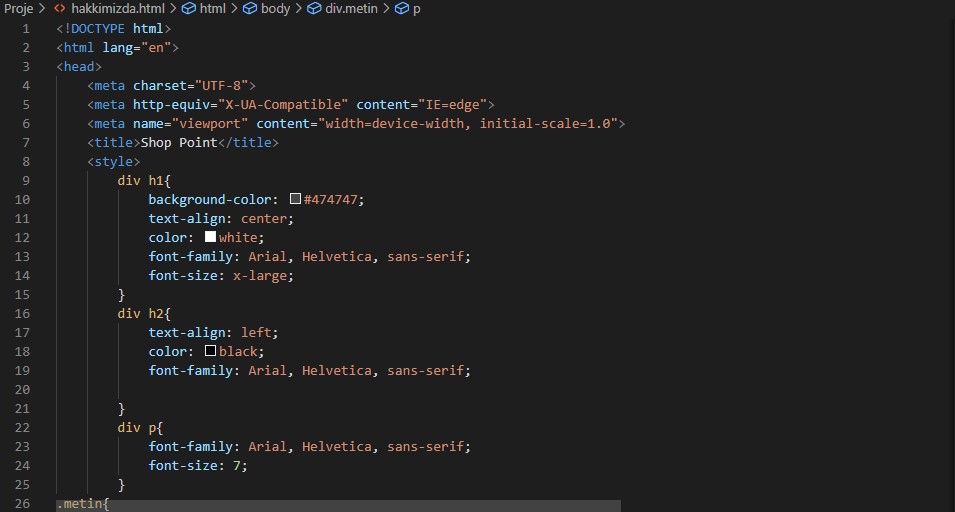


Açıklama: Anasayfayı oluşturan kodlar

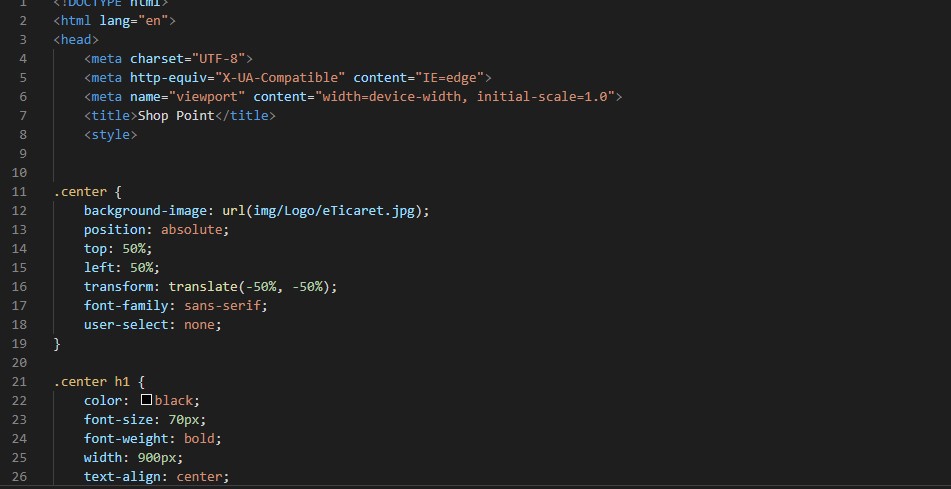
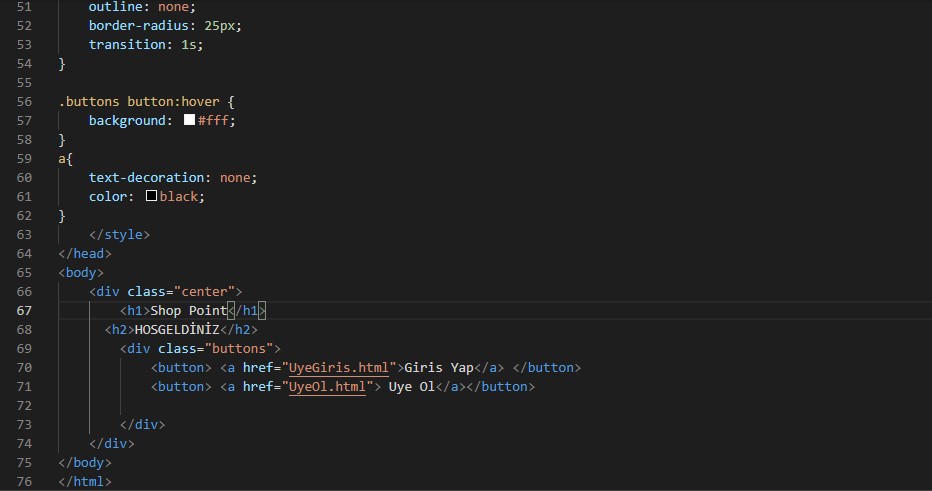
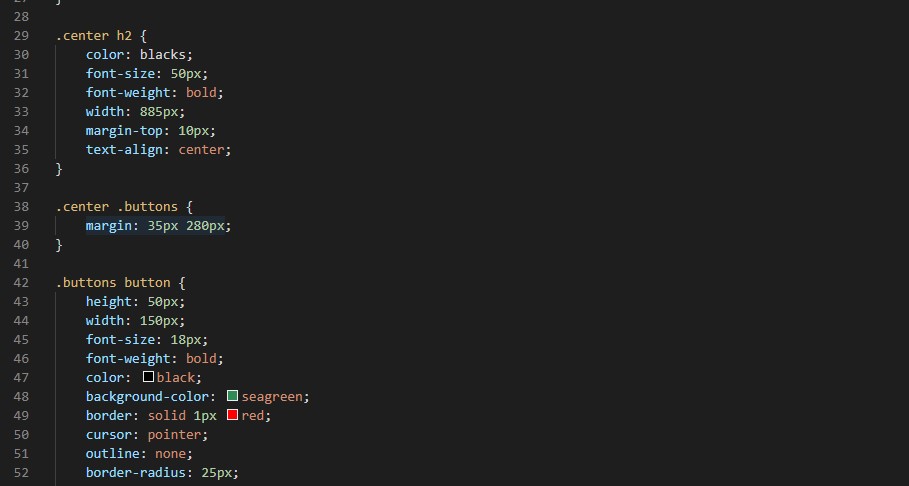




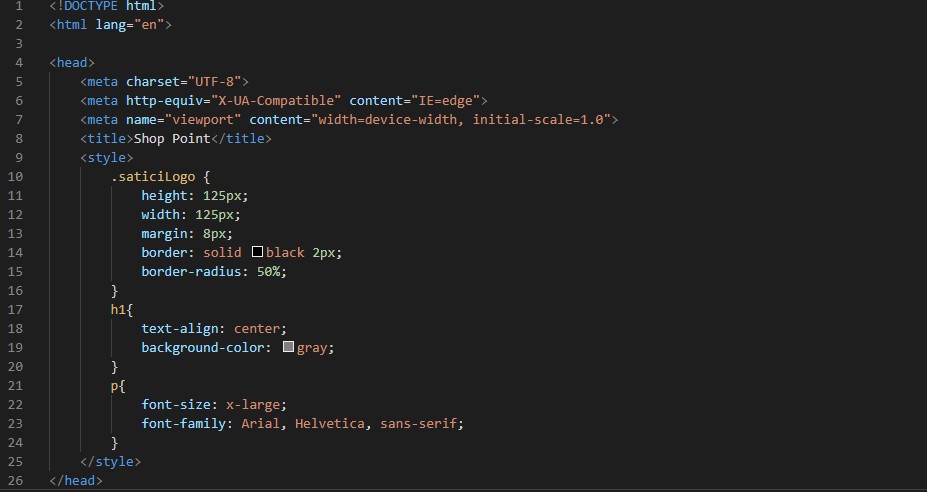
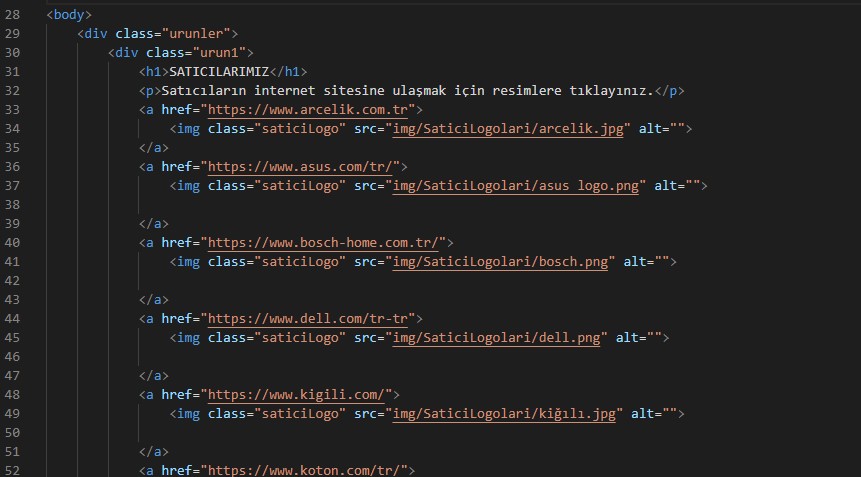
**Açıklama: Navbarda bulunan giris kısmına tıklandığında çıkan arayüz kodları**



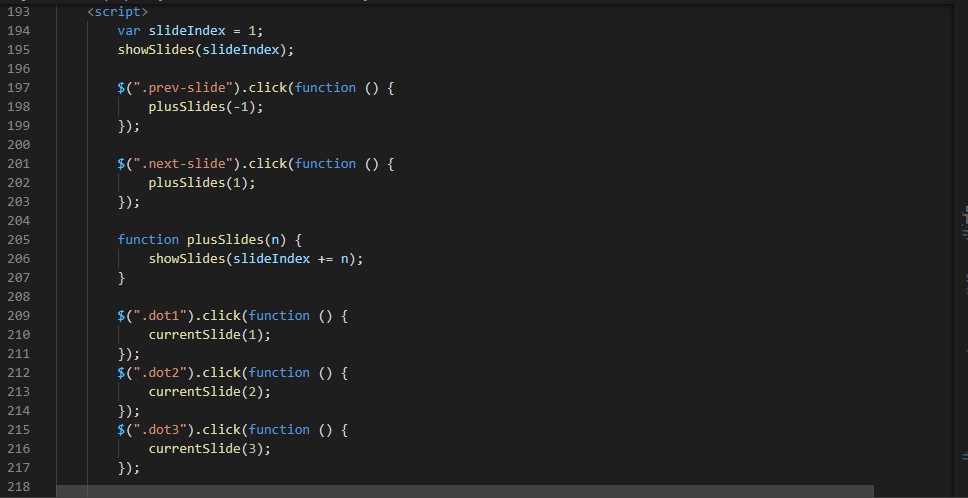
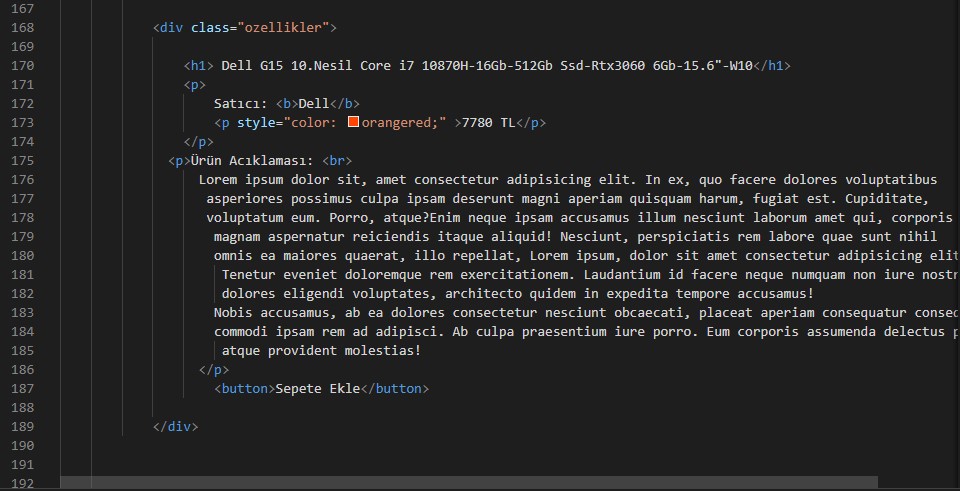
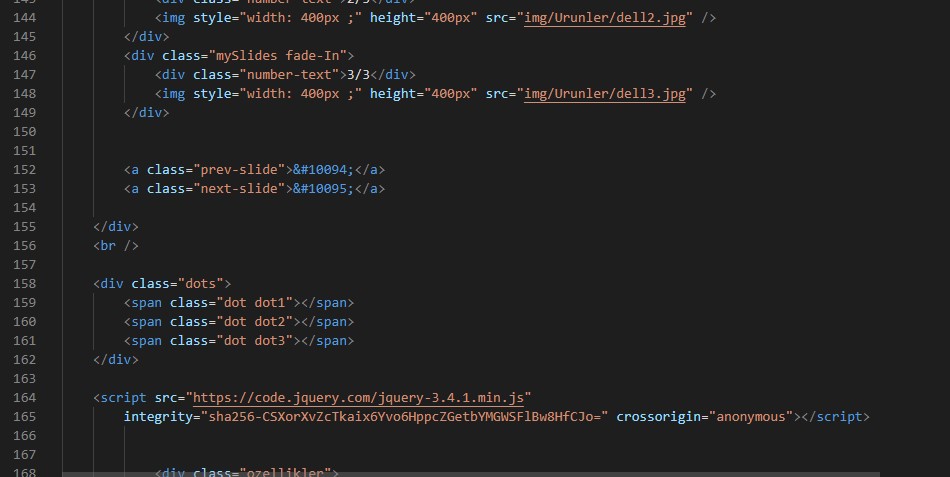
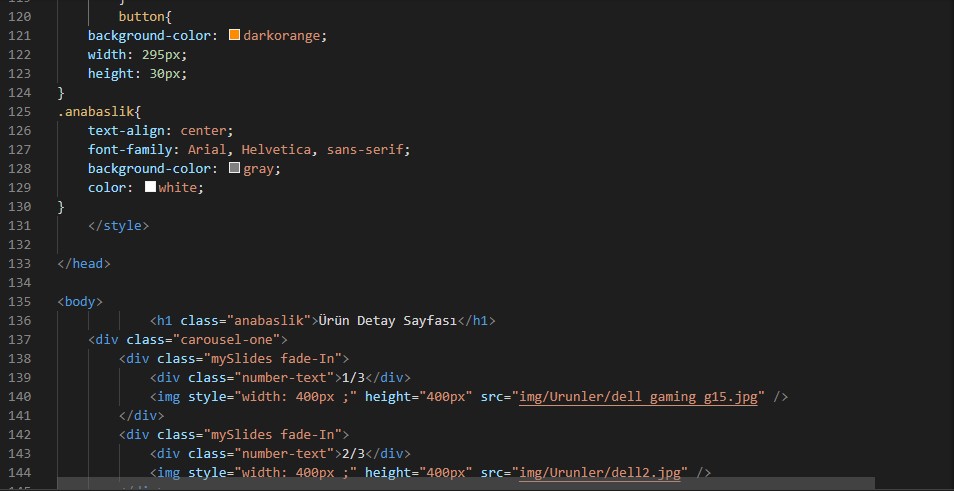
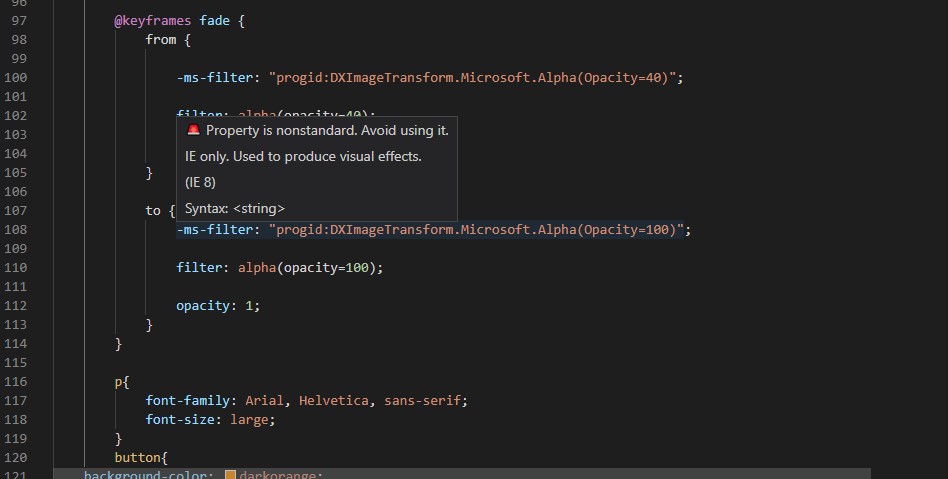
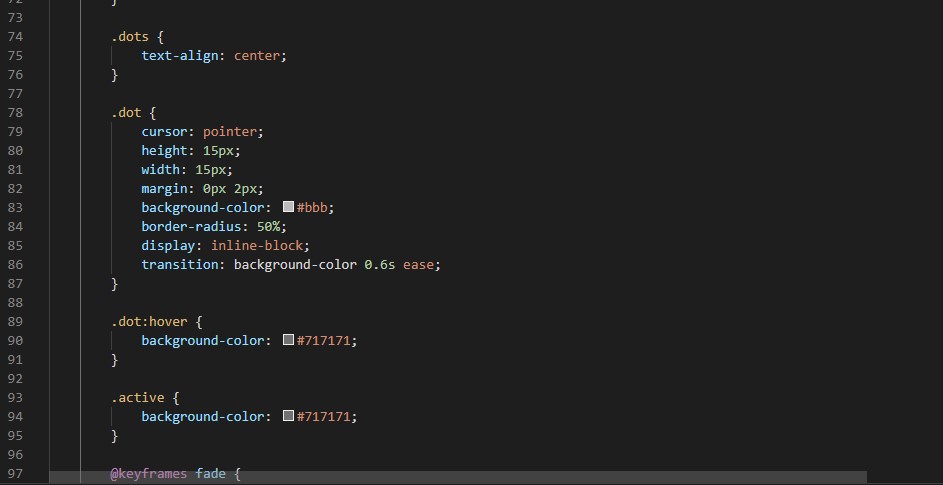
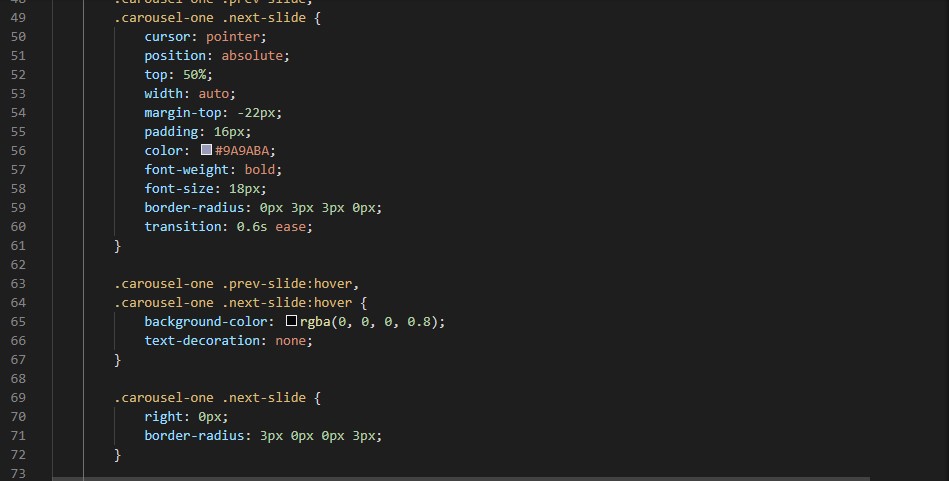
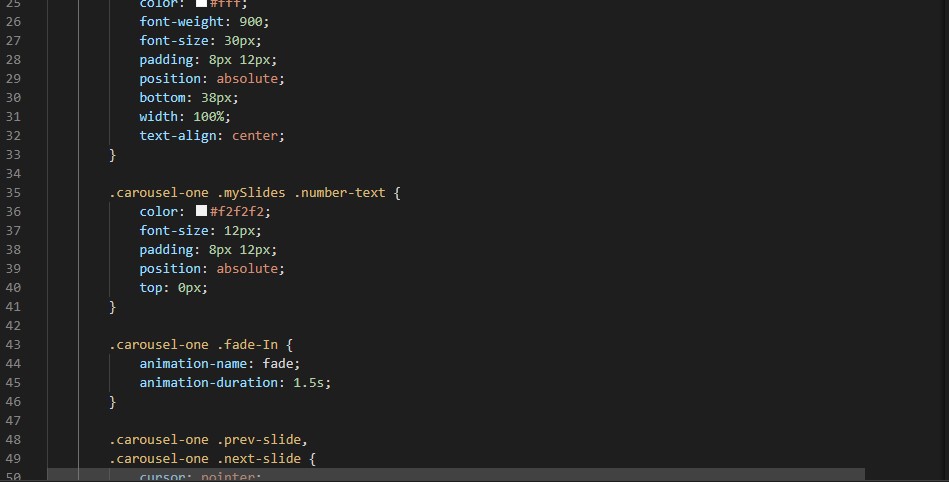
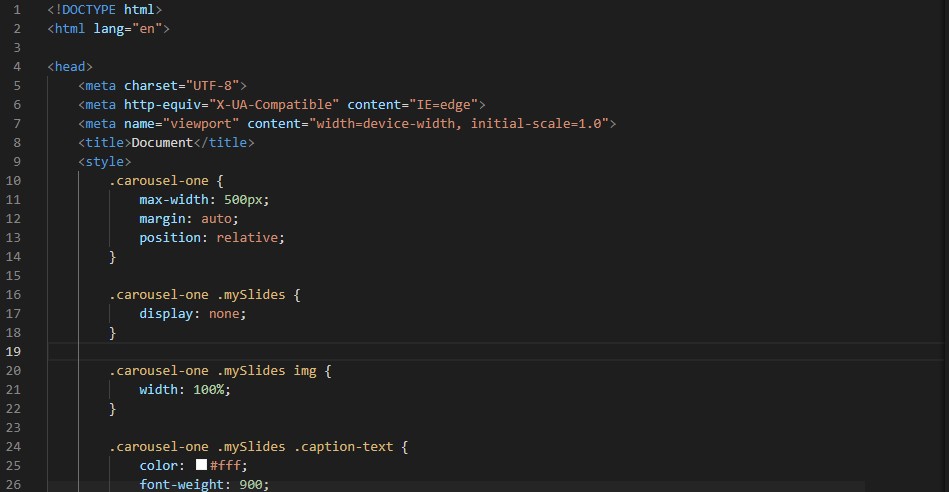
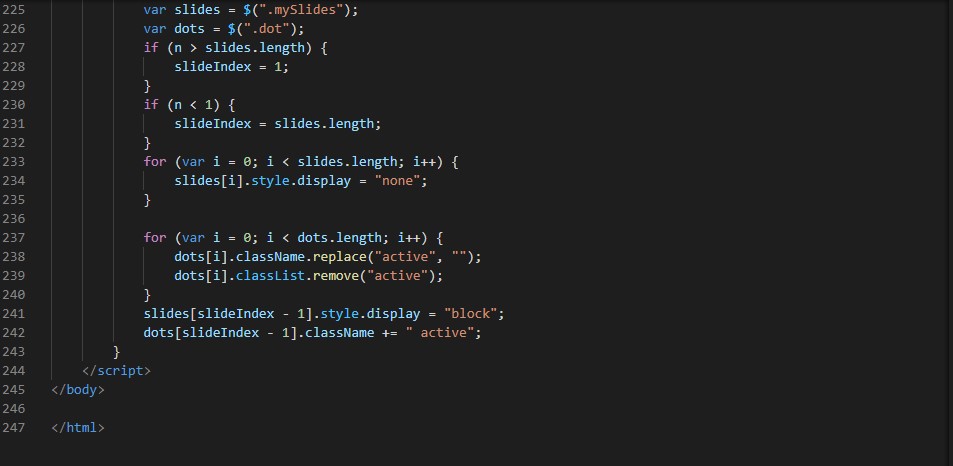
**Açıklama : : Navbarda bulunan hakkımızda kısmına tıklandığında çıkan arayüz kodları**



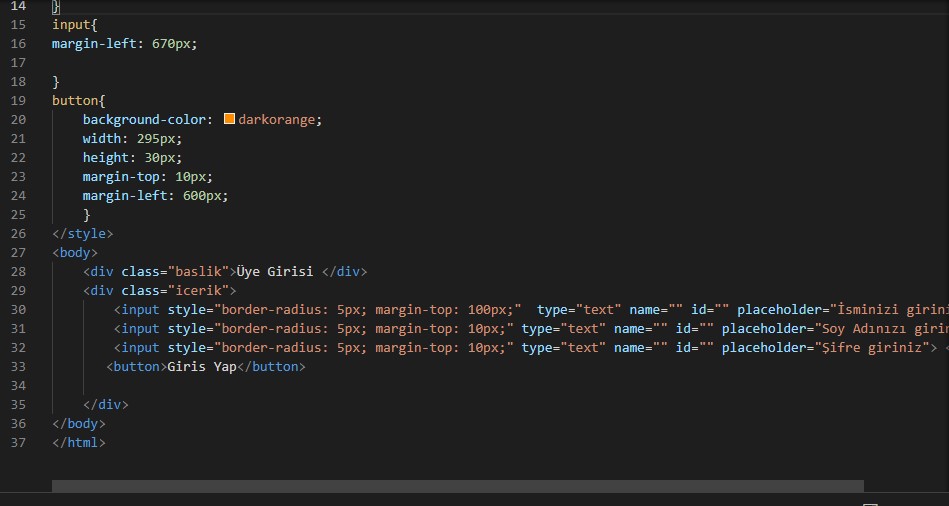
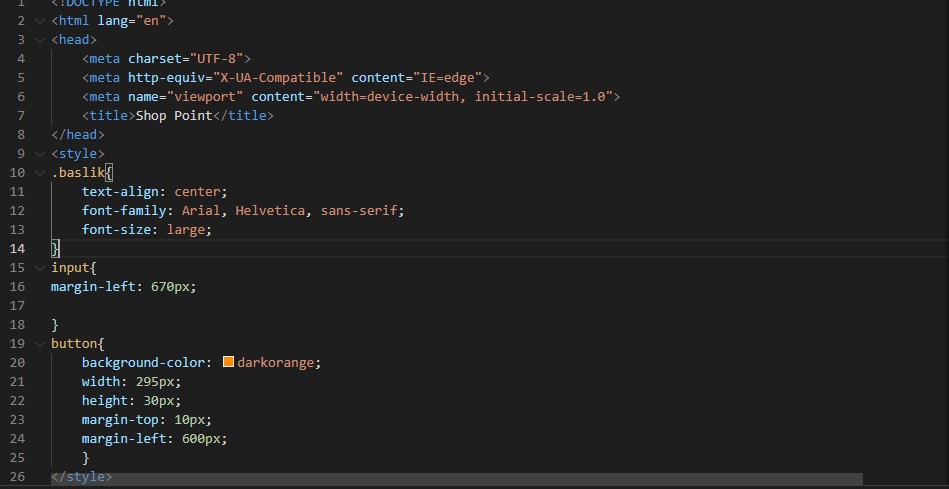
**Açıklama : Navbarda bulunan iletisim kısmına tıklandığında çıkan arayüz kodları**



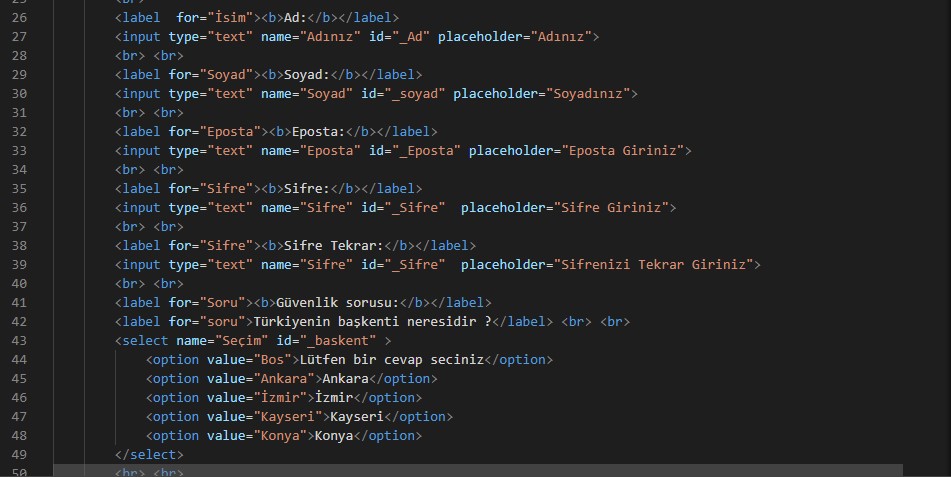
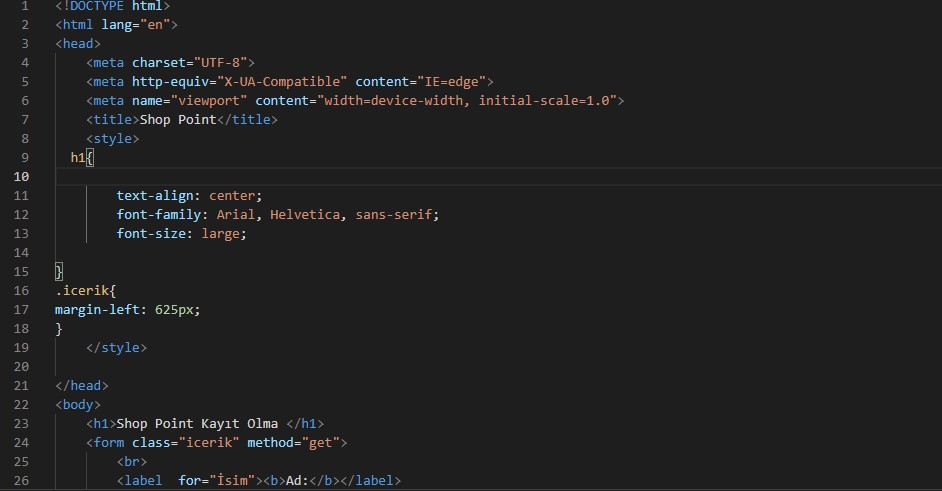
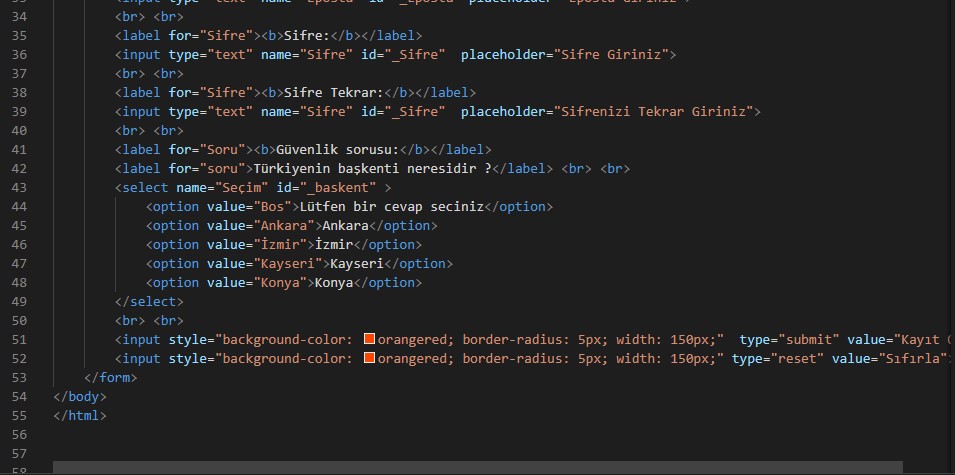
**Açıklama: Navbarda bulunan satıcılar kısmına tıklandığında,satıcıların listelendiği arayüz kodları**



**Açıklama : Anasayfada bulunan ürün resmine tıklandığında gösterilen ürün detay sayfasının kodlarıdır.**



**Açıklama: Sisteme kayıtlı olan üyelerin giriş yaptığı ekranın kodları**



**Açıklama: Sisteme üye olmayan kullanıcıların sisteme kaydedildiği arayüzün kodlarıdır.**

# Ek C: To Be Determined List (TBDL)