

**GAZİ ÜNİVERSİTESİ**

**TEKNOLOJİ FAKÜLTESİ**

**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**BMT-311 VERİ TABANI YÖNETİM SİSTEMLERİ**

**2020-2021 GÜZ DÖNEMİ**

**PROJE RAPORU**

**GRUP ÜYELERİ:**

**<21181616769> – <ALİ SAİT ŞAHİNER>**

**<21181616006> – <ZEKİ FURKAN DURUM>**

## **İÇİNDEKİLER**

[**İÇİNDEKİLER** 2](#_Toc61223217)

[1. GİRİŞ 1-3](#_Toc61223218)

[2. PROJE İÇERİĞİ 4](#_Toc61223219)-17

[2.1. PROJE TANITIMI 4](#_Toc61223220)-7

[2.2. PROJE TASARIM VE GELİŞTİRME AŞAMALARI 7](#_Toc61223221)-15

[2.3. KULLANILAN TEKNOLOJİLER 15-17](#_Toc61223222)

[3. VERİ TABANI TASARIM AŞAMALARI 18-22](#_Toc61223223)

[4. GÜNCEL VERİTABANI VE ER DİYAGRAMI 23-24](#_Toc61223224)

5.KULLANILAN TRİGGER Ve STORED PROCEDURE'LERİN AÇIKLANMASI………………………………………25-38

[6. SONUÇ 38-41](#_Toc61223225)

[7. KAYNAKÇA 42](#_Toc61223226)

1. GİRİŞ

Proje Tanıtımı: Market Otomasyonu ve Stok Takip Sistemi

Bu projemiz, bir market işletmesinin günlük operasyonlarını daha etkili bir şekilde yönetmeyi amaçlayan kapsamlı bir otomasyon ve stok takip sistemini içermektedir. Projemizin ana odak noktası, işletme sahiplerine ve çalışanlarına müşteri yönetimi, stok takibi, satış işlemleri ve genel işletme yönetimini kolaylaştırmak için bir araç sunmaktır.

Projenin Konusu

Proje, bir marketin günlük işlemlerini dijitalleştirerek, müşteri yönetimini ve stok takibini sağlamak üzere tasarlanmış bir otomasyon sistemidir. Bu sistem, hem müşterilere hem de işletme sahiplerine bir dizi özellik sunarak, alışveriş deneyimini iyileştirmeyi ve işletmeyi daha etkili bir şekilde yönetmeyi amaçlamaktadır.

Projenin Amaçları

*1-Verimli Müşteri Yönetimi*

*2-Etkili Stok Takibi*

*3-Satış İşlemlerinin Kolaylaştırılması*

*4-Müşteri ve Ürün Yönetimi*

*5-Marka ve Kategori Yönetimi*

Hedef Kitlesi

*Market İşletmecileri*

İşletme sahipleri, günlük işlemlerini dijitalleştirerek stok takibi, müşteri yönetimi ve satış işlemlerini daha etkili bir şekilde yönetmeyi amaçlayan bu projeden fayda sağlayacaktır.

*Çalışanlar*

Market çalışanları, alışveriş işlemlerini hızlı ve kolay bir şekilde gerçekleştirerek müşterilere daha iyi bir hizmet sunabilirler.

*Müşteriler*

Müşteriler, kişiselleştirilmiş alışveriş deneyimi, sipariş geçmişi takibi ve favori ürünleri kaydetme gibi özelliklerle daha memnuniyetle alışveriş yapabilirler.

Sonuç:

Bu proje, market işletmeleri için daha verimli bir yönetim sistemini sağlayarak, müşterilere daha iyi bir alışveriş deneyimi sunmayı ve işletme sahiplerine de günlük operasyonları daha etkili bir şekilde yönetmeyi hedeflemektedir. Dijitalleşme sayesinde, stok takibi ve müşteri yönetimi gibi süreçler daha hızlı ve hatasız bir şekilde gerçekleştirilebilmektedir.

1. PROJE İÇERİĞİ
   1. PROJE TANITIMI

Projenin Amaçları

*1-Verimli Müşteri Yönetimi*

Müşteriler, kendi hesaplarını oluşturarak geçmiş alışverişlerini inceleyebilir, ürünleri sepete ekleyerek satın alma işlemleri yapabilirler ve kişiselleştirilmiş alışveriş deneyiminden yararlanabilirler.

*2-Etkili Stok Takibi*

Otomasyon sistemi, marketteki ürünlerin stok durumunu anlık olarak takip ederek, stok düzenlemeleri yapmayı ve eğer stokta bir tükenme varsa işletme sahibine bunun uyarısını vermeyi sağlamayı amaçlar.

*3-Satış İşlemlerinin Kolaylaştırılması*

Hem müşteriler hem de işletme sahipleri, kullanıcı dostu bir arayüz üzerinden alışveriş yapabilir ve satış işlemlerini hızlı bir şekilde gerçekleştirebilir.

*4-Müşteri ve Ürün Yönetimi*

Yöneticilerin müşteri ekleyebilir, müşteri bilgilerini güncelleyebilir, müşteri listesini görüntüleyebilir ve aynı şekilde ürün ekleyebilir, ürün bilgilerini güncelleyebilir ve ürün listesini görüntüleyebilir olmasını amaçlar.

*5-Marka ve Kategori Yönetimi*

Yöneticilerin ürünleri kategorilere ve markalara göre düzenleyebilir, yeni kategoriler ve markalar ekleyebilir ve bu sayede ürün yönetimini daha organize bir şekilde gerçekleştirebilir olmasını amaçlar.

Proje Yetenekleri

*1-Veri Tabanı İşlemleri*

PostgreSQL veri tabanı kullanılarak müşteri, ürün, marka, kategori ve stok bilgileri gibi verilerin saklanması ve yönetilmesi.

*2-Kullanıcı Arayüzü Tasarımı*

Windows Forms uygulama arayüzü sayesinde kullanıcı dostu ve anlaşılır bir kullanıcı deneyimi sunulması.

*3-Stok Takip ve Güncelleme*

Ürün stok durumlarının anlık takibi ve stok azalımlarına göre stokları güncelleyebilme veya yeni ürünleri ekleyebilme avantajının sunulması.

*4-Müşteri ve Ürün Yönetimi*

Yeni müşteri ekleme, müşteri silme, mevcut müşteri bilgilerini güncelleme, daha önceden alışveriş yapmış müşterilerin listesini görüntüleme, ürün ekleme, silme, güncelleme ,listeleme işlemlerinin gerçekleştirilmesi.

*5-Satış İşlemleri*

Müşteri sepeti oluşturma, satış işlemlerini kaydetme, stoktan düşme işlemlerini otomatik olarak gerçekleştirme.

Çözümlediğimiz Problemler

*1-Verimsiz İşletme Yönetimi*

Manuel ve kâğıt tabanlı işlemlerle yönetilen işletmelerdeki verimlilik sorunlarına çözüm sunma.

*2-Stok Takip Zorlukları*

Stok düzenlemelerini ve takibini kâğıt üzerinde gerçekleştiren işletmelerin karşılaştığı hataları ve zorlukları aşma.

*3-Müşteri Deneyiminde Zayıflık*

Müşterilere kişiselleştirilmiş bir alışveriş deneyimi sunmayan ve müşteri sadakatini artırmada zorluk yaşayan işletmelerin bu alandaki eksikliklerini giderme.

Çözüm Önerisi ve Yaklaşım

Projenin çözüm önerisi, işletmelerin günlük operasyonlarını dijitalleştirerek daha verimli hale getirmektir. Bu bağlamda aşağıdaki yaklaşımlar sergilenmiştir:

*1-Dijital Dönüşüm*

İşletmelerin kâğıt tabanlı işlemlerini dijitalleştirme, verimliliği artırma ve hata oranını düşürme.

*2-Kullanıcı Dostu Arayüz*

Windows Forms arayüzü ile müşteri ve işletme sahiplerinin rahatlıkla kullanabileceği, anlaşılır ve etkileşimli bir ara yüz tasarımı.

*3-Veri Tabanı Yönetimi*

PostgreSQL veri tabanı kullanarak güvenilir ve ölçeklenebilir bir veri tabanı yönetimi.

*4-Otomasyon ve Gerçek Zamanlı Takip*

Stok takibi, müşteri yönetimi ve satış işlemlerinin otomatikleştirilmesi ve gerçek zamanlı olarak takip edilmesi.

*5- Geniş Kapsamlı Fonksiyonlar*

Müşteri ve ürün yönetimi, stok takibi, satış işlemleri, marka ve kategori düzenlemeleri gibi geniş bir yelpazede işlevselliğe sahip bir otomasyon sistemi.

* 1. PROJE TASARIM VE GELİŞTİRME AŞAMALARI

Tasarım Aşaması

*Kullanıcı Araştırması*

Kullanıcıların ihtiyaçlarını belirlemek ve kullanıcı dostu bir arayüz tasarlamak adına internet üzerinden bir pazar araştırması gerçekleştirdik. İşletmelerin nasıl bir programa ihtiyaçları olduğunu bu araştırmanın sonucunda planladık ve kabaca tasarımını yaptık. Hem müşteri hem de işyeri sahiplerinin kullanımına uygun entegre bir programı nasıl yapabileceğimizi planladık. İşletme sahipleri, çalışanlar ve müşterilerin kullanıcı deneyimini en üst düzeye çıkarmayı amaçladık.

*Ekran Planlaması ve Prototip*

Belirlenen ihtiyaçlar doğrultusunda modelleme ve ekran planlamaları oluşturarak, uygulamanın temel tasarımını çıkardık. Prototip üzerinde yapılan kullanılabilirlik testleri sayesinde tasarım hatalarını önceden tespit edip düzeltme şansı elde ettik.

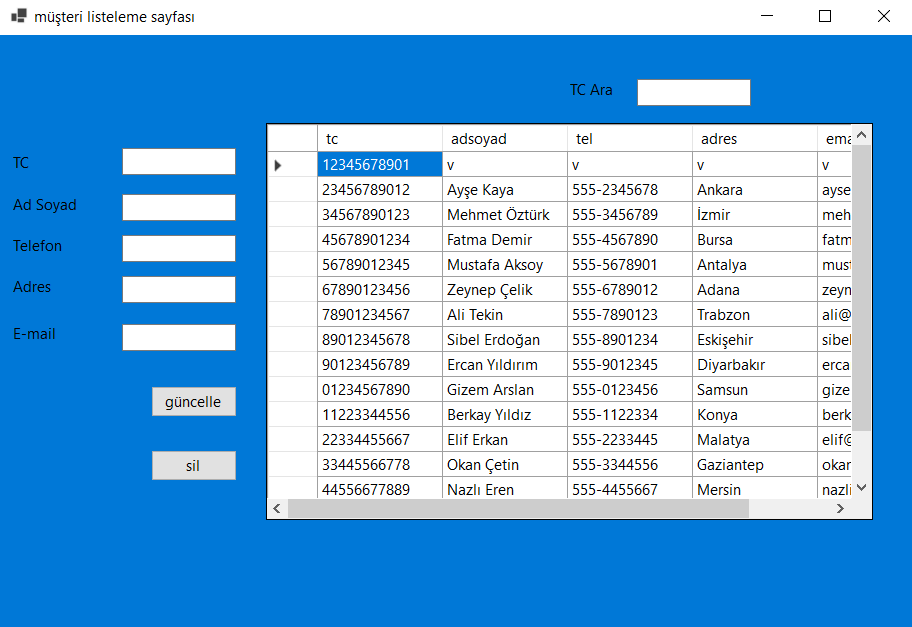
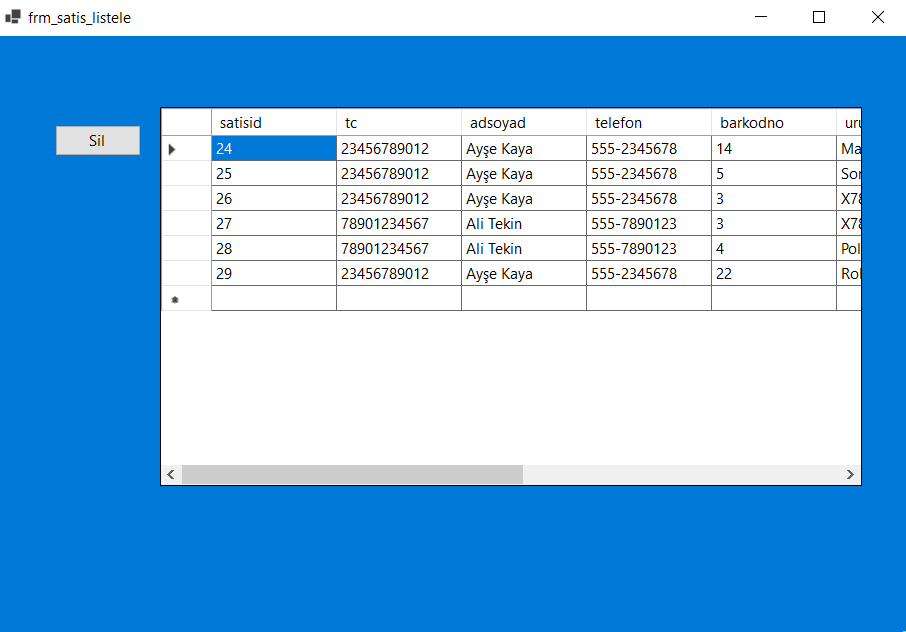
*Tema ve Renk Seçimi*

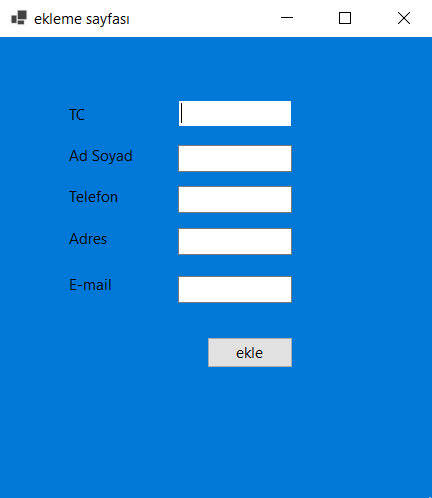
Uygulamanın renk paletini, işletmenin kurumsal kimliğine uygun olarak belirledik. Uygulamamıza açık ve orta tonlarda mavi renklerinin kullanımını tercih ettik. Buton seçimlerini, kullanıcıların hızlı bir şekilde anlayabileceği ve etkileşimde bulunabileceği öğeleri seçerek gerçekleştirdik.

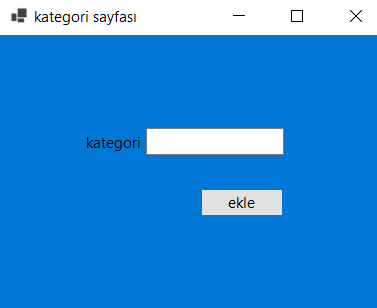
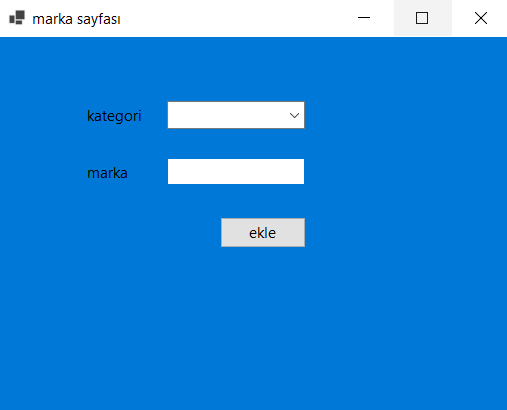
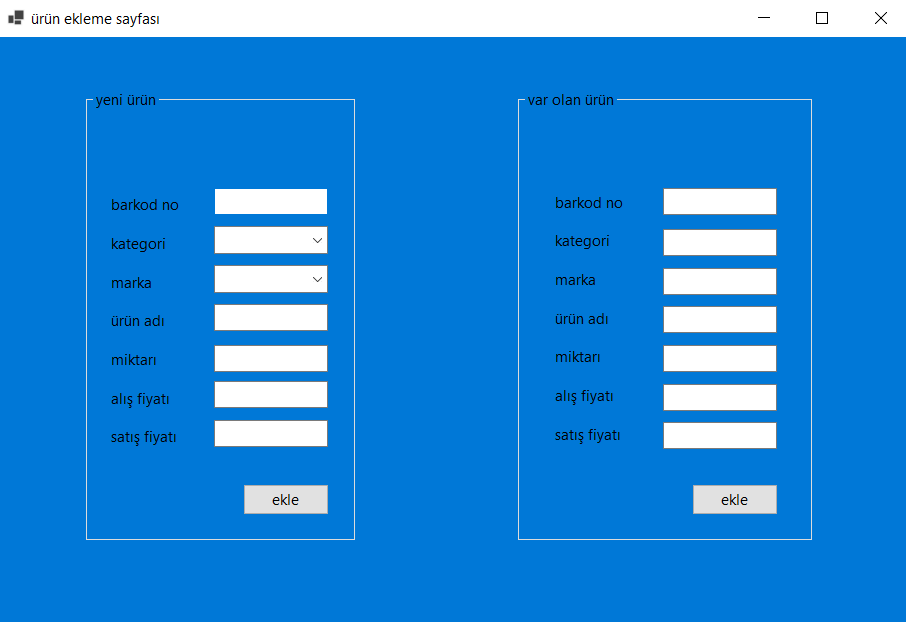
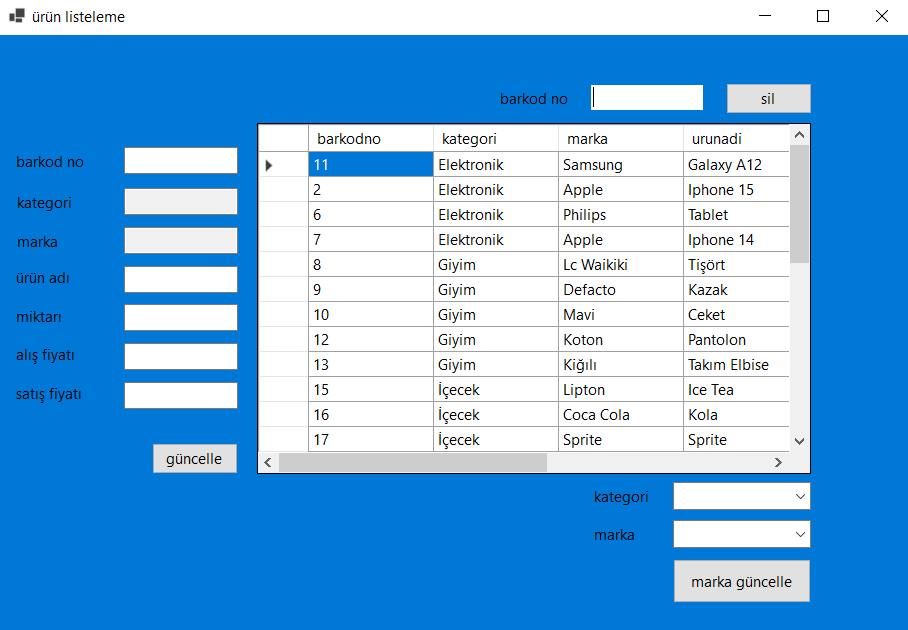
*Login Ekranı Tasarımı*

Kullanıcıların ve işletme sahiplerinin sisteme giriş yapabileceği bir login ekranı tasarladık, kullanıcı adı ve şifre girişi için uygun alanlar oluşturulduk. Görsel ve metin öğeleri kullanarak kullanıcıyı bilgilendiren bir tasarım hedefledik.

WİNDOWS FORM TASARIM ARAYÜZLERİmetin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazılım, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu



Geliştirme Aşaması

*Veri Tabanı Yapısı*

PostgreSQL veri tabanını tercih ettik ve veri tabanı yapısını, müşteri bilgileri, uygulamaya giriş bilgileri, ürün bilgileri, satış işlemleri , marka bilgileri, kategori bilgileri gibi entegre bir şekilde tasarladık.

İlişkisel veri tabanı modellemesi yaparak, veri bütünlüğünü sağladık.

*Arayüz Geliştirme*

Windows Forms uygulaması olarak geliştirdik. Arayüz tasarımında kullanıcı dostu bir deneyim sunabilmek adına kontrol elemanlarını (TextBox, Button, DataGridView, ComboBox vb.) düzenli ve anlaşılır bir şekilde yerleştirdik.

Arayüzdeki butonları, işlevselliklerine göre gruplandırıldık.

*Sepet İşlemleri*

Müşterinin alışveriş yaparken sepete ürün ekleyebilmesi, çıkartabilmesi ve satış işlemini tamamlayabilmesi adına özel bir sepet sınıfı oluşturduk.

Sepet işlemlerini yönetmek için bu sınıfı kullanarak, müşteri sepetini düzenleyebilme yeteneği ekledik.

*Müşteri ve Ürün Yönetimi*

Müşteri ekleme, silme, güncelleme ve listeleme işlemleri için ayrı form sayfaları tasarladık.

Ürün ekleme, silme, güncelleme ve listeleme işlemleri için ayrı form sayfaları tasarladık.

Müşteri ve ürün listelerini DataGridView kontrolü ile kullanıcıya gösterdik.

*Hata Yönetimi ve İstisna Durumları*

Hataların kullanıcı dostu bir şekilde yönetilmesi adına try-catch blokları ekledik.

İstisna durumları yöneterek, kullanıcıya anlamlı hata mesajları sunmaya özen gösterdik.

*Veri tabanı İşlemleri:*

İlk adım olarak, veritabanı tasarlandı ve normalizasyon işlemleri uygulandı. Veri tabloları arasındaki ilişkiler kurularak, veri bütünlüğü sağlandı.

Normalizasyon işlemleri, veritabanındaki verilerin tutarlılığını artırmak ve gereksiz tekrarları önlemek adına yapıldı. Daha sonrasında aşağıdaki maddeler veri tabanı tasarımına eklendi.

*1-SQL Kütüphanesi seçimi:*

Npgsql kütüphanesini kullanarak PostgreSQL veri tabanına bağlandık.

Parametreli sorgular ile güvenli bir veri tabanı erişimi sağladık.

*2-Tablo Oluşturma:*

Veri tabanımızda kullanmayı planladığımız tablolar oluşturuldu. Tabloların içeriği gerekli sütunlarla dolduruldu ve tabloların Primary Key’leri belirlendi.

*3-Stored Procedure’ler:*

Veri tabanında gerçekleştirilecek işlemleri kolaylaştırmak ve güvenliğini artırmak amacıyla stored procedures oluşturuldu.

Özellikle kullanıcı ekleme, silme, güncelleme işlemleri için stored procedures kullanıldı.

*4-Trigger’lar*

Veritabanında belirli bir olay gerçekleştiğinde tetiklenen triggerlar kullanıldı.

Örneğin, satış işlemi gerçekleştiğinde stoktan düşme ve negatif stok seviyesine düşüldüğünde uyarı verme gibi senaryolarda triggerlar devreye girdi.

Tasarım Aşamasında Karşılaşılan Problemler ve Çözümler

*Yetkilendirme ve Güvenlik*

Problem: Admin ve müşteri rollerinin doğru şekilde yönetilmesi ve kullanıcılara sadece gerekli olan işlemleri yapma yetkisi verilmesi gerekiyordu.

Çözüm: Kodlama ile bu problemin üstesinden geldik bu sayede 2 adet kullanıcı tipimiz oluşmuş oldu.

*Normalizasyon ve Performans*

Problem: Veri tabanındaki tabloların normalizasyonu sırasında, performans kayıplarını minimize etmek adına doğru dengeyi bulmak zor oldu.

Çözüm: Normalizasyon işlemleri yapıldı. Bu sayede veri tabanı sorgularının daha hızlı çalışması sağlandı.

Geliştirme Aşamasında Karşılaşılan Problemler ve Çözümler

*Ara Yüz Tasarımı ve Kullanılabilirlik*

Problem: Müşteri ve admin arayüzlerinin kullanıcı dostu olması ve kolay anlaşılabilir olması gerekiyordu.

Çözüm: Ara yüz tasarımında kullanıcı dostu bir yaklaşım benimsendi. Özellikle müşteri tarafında alışveriş sürecinin basit ve hızlı olabilmesi için minimalist bir tasarım tercih edildi. Admin tarafında ise yönetim işlemlerini rahatça gerçekleştirebilmek adına detaylı ve organize bir arayüz oluşturuldu.

*Trigger ve Stored Procedures Kullanımı*

Problem: Veri tabanı işlemlerinin daha güvenli ve performanslı olabilmesi için doğru triggerlar ve stored procedures kullanılmalıydı.

Çözüm: PostgreSQL üzerinde uygun triggerlar ve stored procedures oluşturularak, özellikle satış işlemleri ve stok kontrolü gibi işlemlerde veri tabanı güvenliği ve performansı artırıldı.

*Normalizasyon ve Veri Bütünlüğü*

Problem: Normalizasyon sırasında bazı tablolar arasındaki ilişkilerin karmaşıklığı, veri bütünlüğünü sağlamakta zorluklar doğurdu.

Çözüm: Yapılan normalizasyon işlemleri ile veri bütünlüğü sağlanmaya çalışıldı. Yapılan bu işlemler ile veri tekrarlarının önüne geçilmeye çalışıldı, ilişkisel bütünlük daha da güçlendirildi.

* 1. KULLANILAN TEKNOLOJİLER

Kullandığımız Teknolojiler:

*C# Programlama Dili:*

Neden Kullanıldı? C#, nesne yönelimli programlama (OOP) yapısına sahip, güçlü ve esnek bir programlama dilidir. Windows tabanlı masaüstü uygulamalar geliştirmek için uygun bir seçenek olması, dilin geniş kütüphane desteği ve kolay öğrenilebilir olması nedeniyle ve grup üyeleri olarak her ikimizin de bu dili bilmesi nedeniyle tercih edildi.

*Windows Forms (WinForms):*

Neden Kullanıldı? Windows Forms, C# ile entegre bir şekilde çalışan bir kullanıcı arayüzü (UI) framework'üdür. Görsel olarak zengin ve kullanımı kolay arayüzler oluşturmak için tercih edildi. Form tasarımı için sürükle-bırak yöntemi, hızlı bir şekilde arayüz oluşturmayı sağladı. Kullanıcı dostu bir arayüz tasarlamamız için gereken bütün özellikler mevcut olduğundan bu framework tercih edildi.

*PostgreSQL Veritabanı:*

Neden Kullanıldı? PostgreSQL, açık kaynaklı bir ilişkisel veri tabanı yönetim sistemidir. Geniş veri tipi desteği ve güvenilirliği ile bilinen bir veri tabanıdır. Halihazırda bu dönem laboratuvarda kullandığımız veri tabanı bu olduğundan dolayı projenin veri tabanı yönetimi için tercih edildi.

*Visual Studio IDE:*

Neden Kullanıldı? Visual Studio, C# ve .NET tabanlı projelerin geliştirilmesi için güçlü bir entegre geliştirme ortamıdır. Hata ayıklama, otomatik tamamlama, arayüz tasarımı gibi özellikleriyle geliştirme sürecini hızlandırdı.

*Npgsql (PostgreSQL için .NET Sürücüsü):*

Neden Kullanıldı? Npgsql, C# ile PostgreSQL veri tabanına erişim sağlayan bir .NET sürücüsüdür. Bu kütüphane sayesinde C# uygulamamızın PostgreSQL veri tabanı ile etkileşimi kolay ve verimli bir şekilde gerçekleşti.

*Draw.io ve Powerpoint:*

Neden Kullanıldı? Bu araçlar, proje tasarımı sırasında kullanılan görsel tasarım araçlarıdır. Veri tabanı tasarımı, UML ve ER diyagramlarının oluşturulması , kullanıcı arayüzü planlaması ve diğer mimari kararların belgelenmesinde yardımcı oldular.

1. VERİ TABANI TASARIM AŞAMALARI

VARLIK TİPLERİNİ OLUŞTURMA:

*1-SEPET*

Açıklama : Müşterinin seçtiği ürünleri bir arada tutan güçlü bir varlıktır. Primary Key’i sepet’dir.

*2-ÜRÜN*

Açıklama : Markette bulunan ürünlerin özelliklerini içinde barındıran güçlü bir varlıktır. Primary Key’i barkodNo ‘dur.

*3-MÜŞTERİ*

Açıklama: Marketten alışveriş yapan müşterilerin bilgilerini içinde barındıran güçlü bir varlıktır. Primary Key’i kimlik numarası bilgisini tutan tc ‘ dir.

*4-SATIŞ*

Sepet işlemlerinden sonra eğer sepetteki ürünler satın alınılmaya karar verilirse bu ürünlerin bilgisini içinde tutan güçlü bir varlıktır. Primary Key’i satis\_id’dir.

*5-MARKA BİLGİLERİ*

Açıklama: Ürünlerin hangi markanın ürünü olduğunu belirten güçlü bir varlıktır. Primary Key’i marka\_id’dir.

*6-KATEGORİ BİLGİLERİ*

Açıklama : Ürünlerin reyonlarda hangi kategorilere göre ayrılacağını belirten güçlü bir varlıktır.Primary Key’i kategori\_id ‘dir.

diyagram, daire, taslak, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

diyagram, daire, yazı tipi, plan içeren bir resim

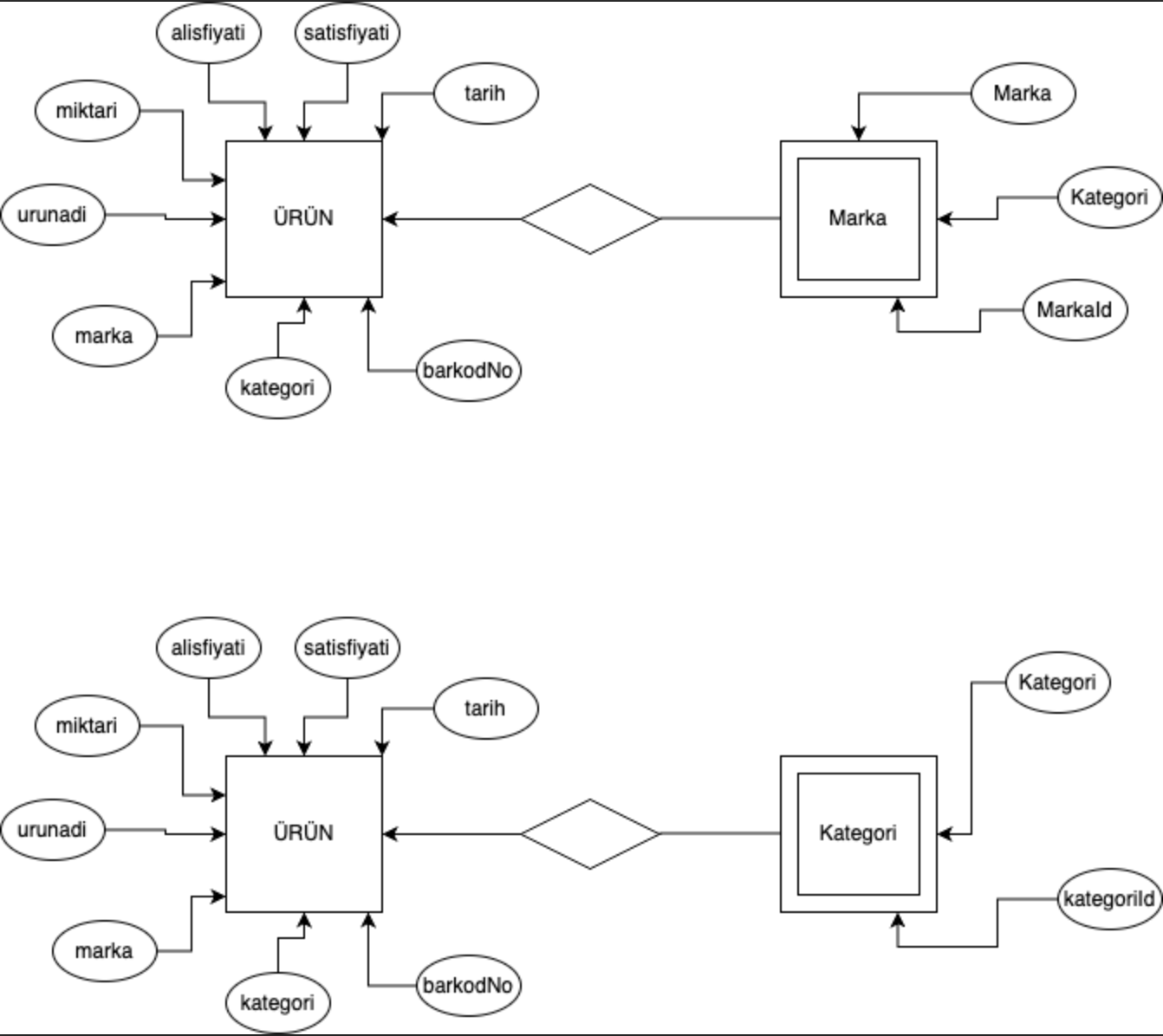
Açıklama otomatik olarak oluşturuldu

diyagram, taslak, daire, plan içeren bir resim

Açıklama otomatik olarak oluşturuldu

diyagram, taslak, daire, çizim içeren bir resim

Açıklama otomatik olarak oluşturuldu



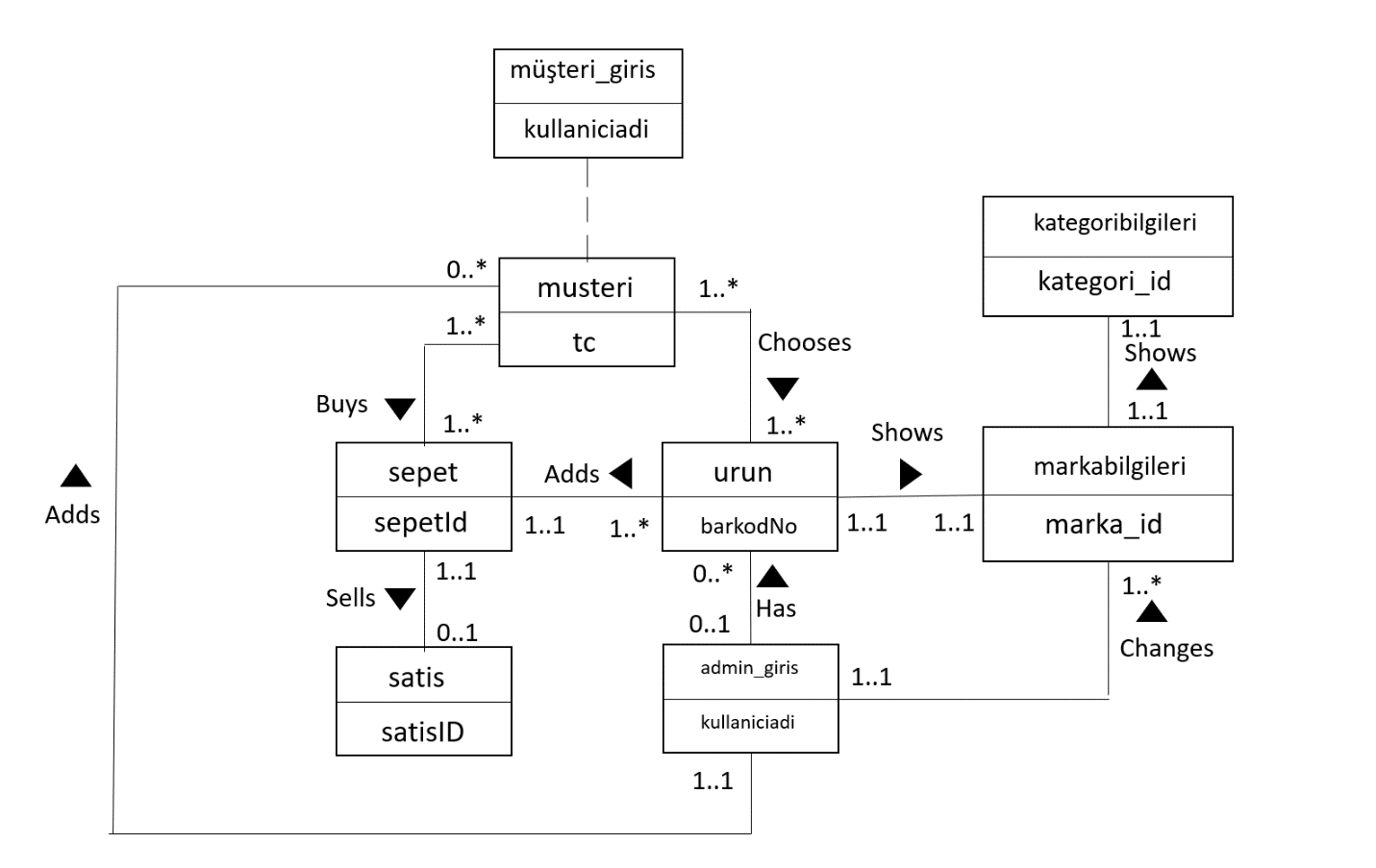
1. GÜNCEL VERİTABANI VE ER DİYAGRAMI

GÜNCEL VERİTABANI

metin, diyagram, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

ENTİTY RELATİONSHİP DİYAGRAMI



1. KULLANILAN TRIGGER VE STORED PROCEDURE’LERİN AÇIKLANMASI

PROSEDÜRLER

-MÜŞTERİ EKLE

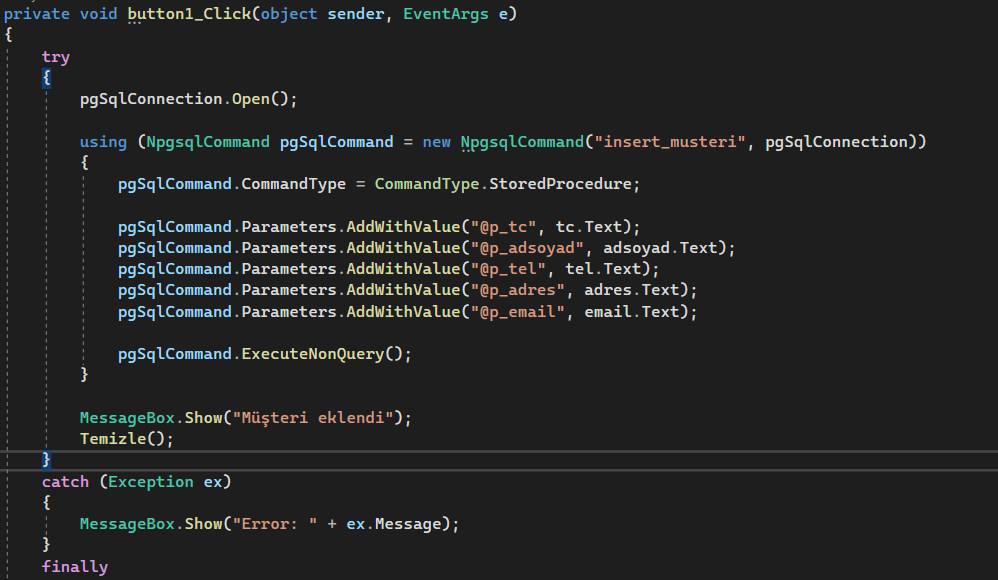
*-PROSEDÜR TANIMI*

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

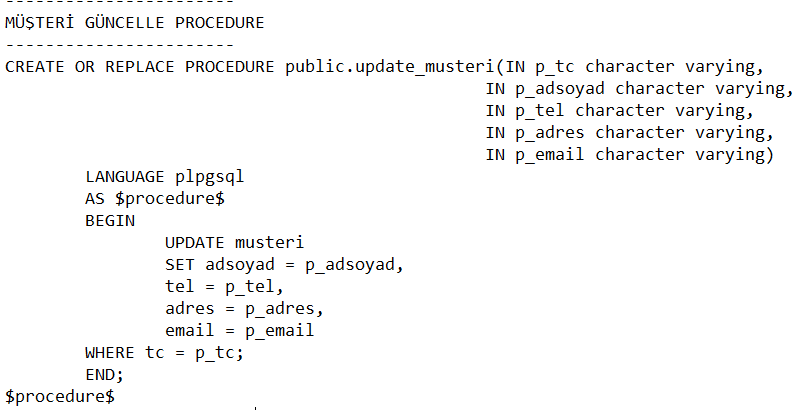
*-PROSEDÜR AÇIKLAMASI:*

Parametre olarak character tipindep\_tc,p\_adsoyad,p\_tel,p\_adres,p\_email değerlerini alır ve bu değerleri müşteri tablosuna ekler.

-*PROSEDÜR ÇALIŞTIRILMASI*

-MÜŞTERİ GÜNCELLE

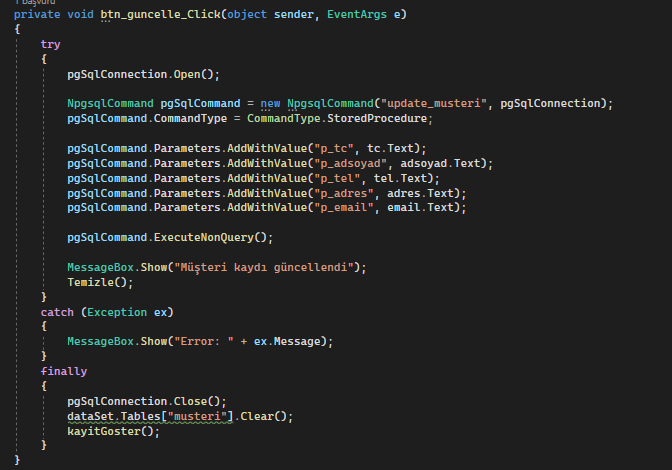
*-PROSEDÜR TANIMI*



*PROSEDÜR AÇIKLANMASI:*

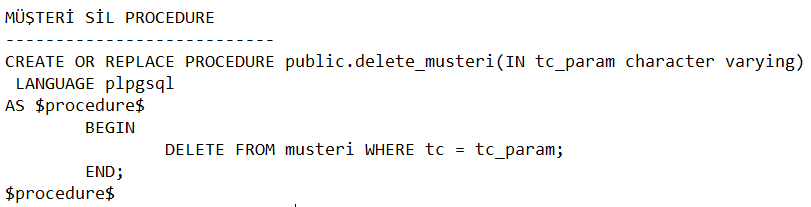
Parametre olarak p\_tc,p\_adsoyad,p\_tel,p\_adres,p\_email parametrelerini alır ve bu parametlereri müşteri tablosundaki eşleşen tc numarasının eski müşteri bilgileriyle değiştirir.

*-PROSEDÜR ÇALIŞTIRILMASI*



-MÜŞTERİ SİL

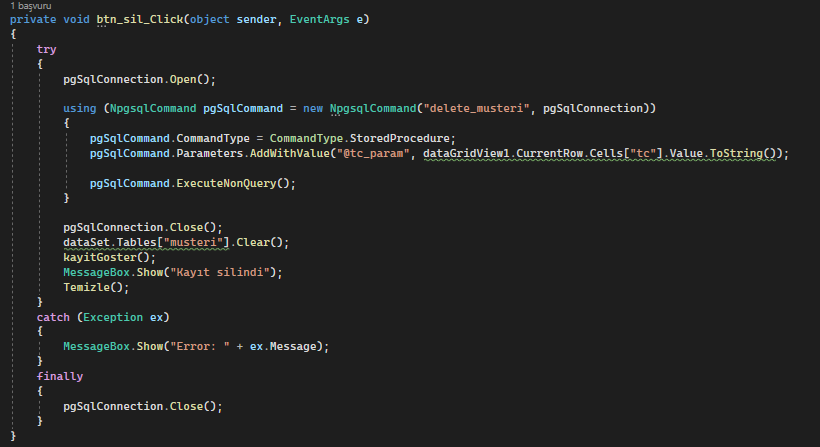
*-PROSEDÜR TANIMI*



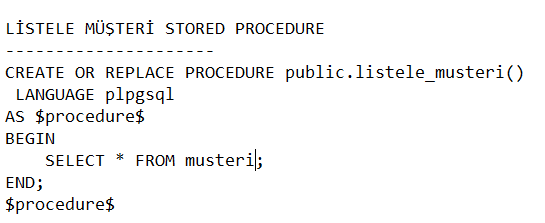
*-PROSEDÜR AÇIKLAMASI:*

Parametre olarak tc\_param alır ve bu parametretle eşleşen tc kimlikli müşteriyi tablodan siler.

*-PROSEDÜR ÇALIŞTIRILMASI*



-MÜŞTERİ LİSTELE

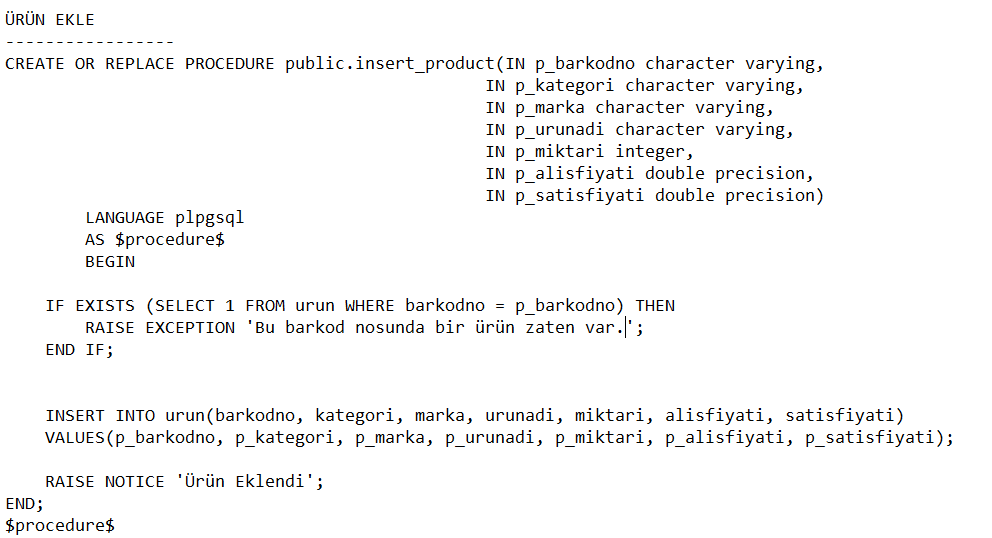


*-PROSEDÜR AÇIKLAMASI:*

Müşteri tablosundaki tüm değerleri listeler.

-ÜRÜN EKLE

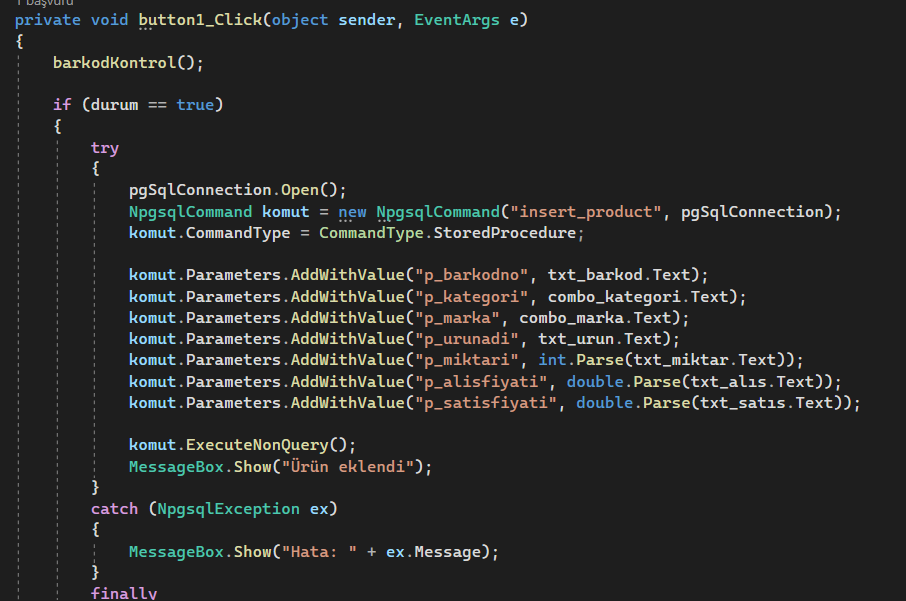
*-PROSEDÜR TANIMLAMASI*



*-PROSEDÜR AÇIKLAMASI:*

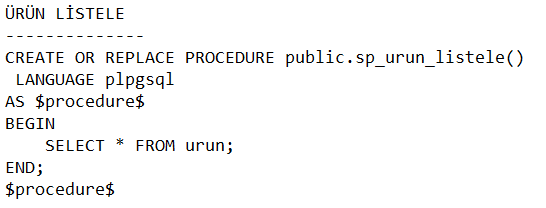
Parametre olarak p\_barkodno,p\_kategori,p\_marka,p\_urunadi,p\_miktari,p\_alisfiyati,p\_satisfiyati değerlerini alır. Eğer p\_barkodno ile eşleşen bardok nolu bir ürün varsa uyarı verir. Yoksa parametre olarak verilen değerler yeni bir satır olarak ürün tablosuna eklenir ve ‘Ürün eklendi.’ mesajı verilir.

*-PROSEDÜR KULLANIMI*



-ÜRÜN LİSTELE

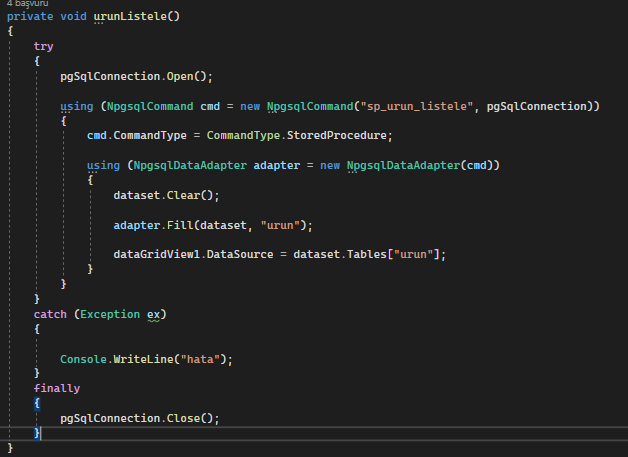
*-PROSEDÜR TANIMLANMASI*

-

*-PROSEDÜR AÇIKLAMASI*

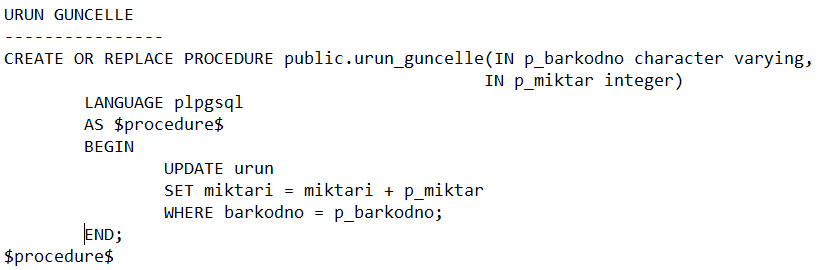
Ürün tablosundaki tüm değerleri listeler.

*-PROSEDÜRÜN ÇALIŞTIRILMASI*



-ÜRÜN GÜNCELLE

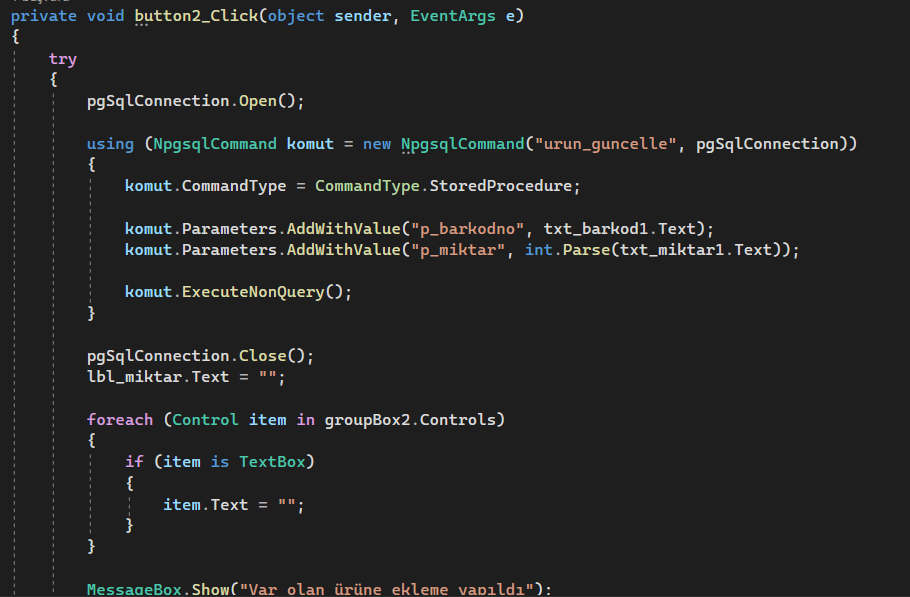
*-PROSEDÜR TANIMI*



*-PROSEDÜR AÇIKLAMASI :*

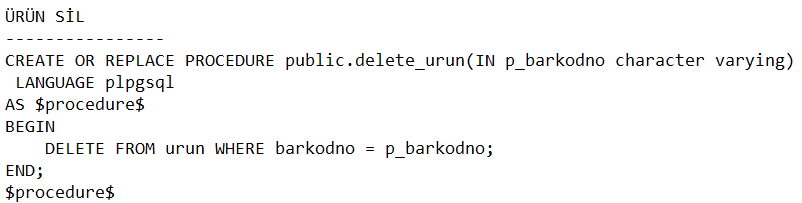
Parametre olarak p\_barkodno ve p\_miktar alır. Bardok no ile eşleşen ürünün miktarını günceller.

*-PROSEDÜRÜN ÇALIŞTIRILMASI*



-ÜRÜN SİL

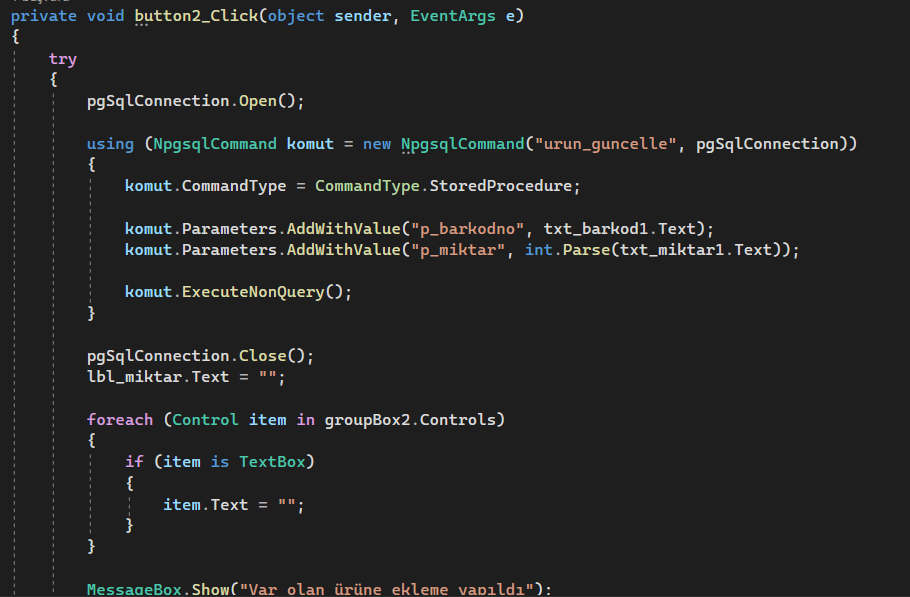
*-PROSEDÜRÜN TANIMI*



*-PROSEDÜRÜN AÇIKLANMASI:*

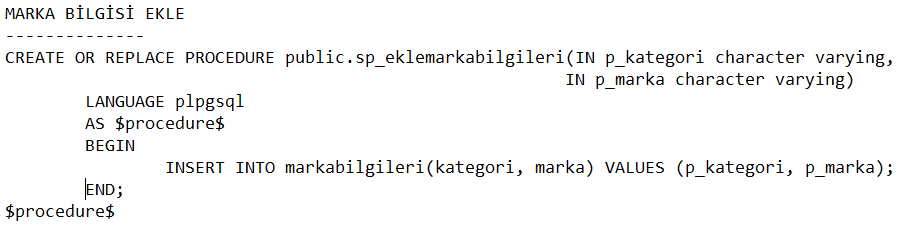
Parametre olarak p\_barkodno alır ve bu barkod no ile eşleşen barkod nolu ürünü tablodan siler.

*-PROSEDÜRÜN ÇALIŞTIRILMASI*



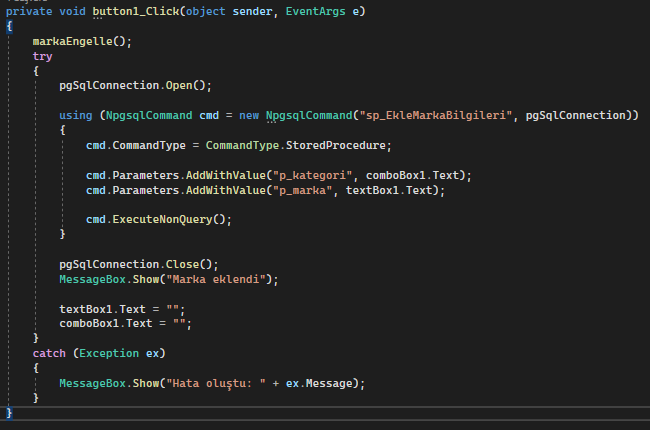
-MARKA BİLGİSİ EKLE

*-PROSEDÜRÜN TANIMLANMASI*



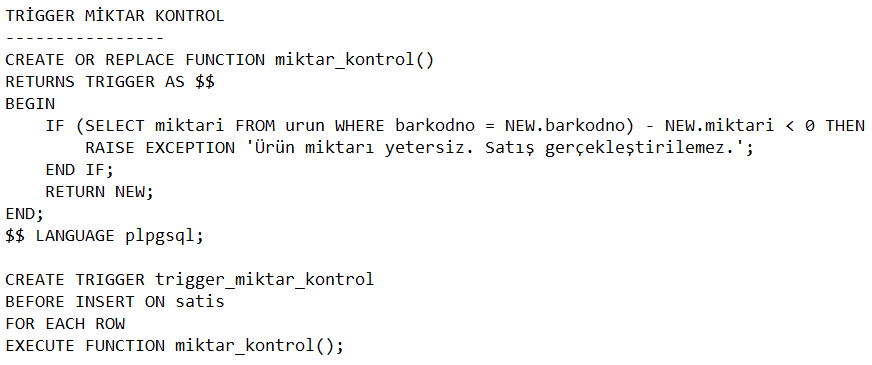
*-PROSEDÜRÜN AÇIKLANMASI:* Parametre olarak p\_kategori ve p\_marka alır. Bu bilgileri markabilgileri tablosuna ekler.

*-PROSEDÜRÜN KULLANILMASI*



TRİGGERLAR

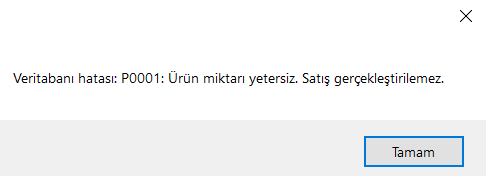
-TRİGGER NEGATİF MİKTAR KONTROL TANIMLANMASI



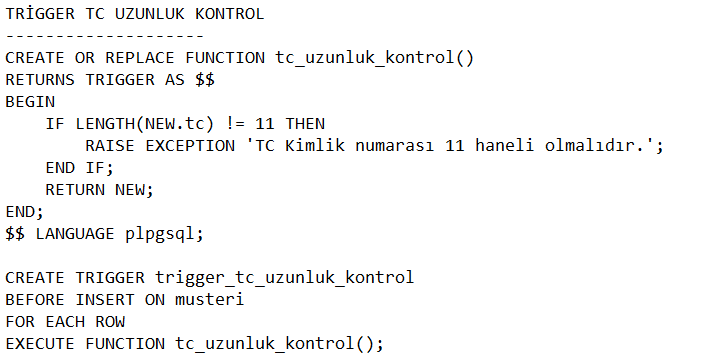
*-TRİGGER AÇIKLAMASI:*

Eğer satışı yapılacak barkod nolu ürünün miktarı 0’dan küçük olursa uyarı fırlatır. Bu trigger satış tablosuna bir kayıt eklemeden önce çalışır.

*-TRİGGER NEGATİF MİKTAR KONTROL ÇALIŞTIRILMASI*



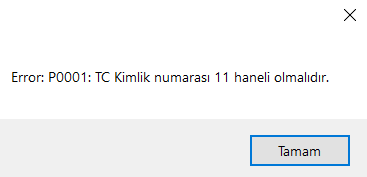
-TRİGGER TC UZUNLUK KONTROL TANIMLANMASI



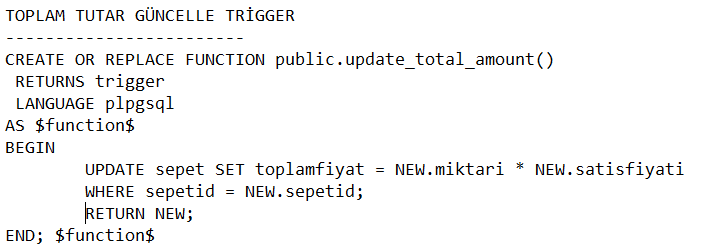
*-TRİGGER AÇIKLAMASI:*

Eğer yeni girilien tc kimlik numarasının uzunluğu 11 haneli değilse hata fırlatır. Müşteri tablosuna eklenmeden önce çalışır.

*-TRİGGER TC UZUNLUK KONTROL ÇALIŞTIRILMASI*



-TRİGGER TOPLAM TUTAR GÜNCELLE TANIMLANMASI



*-TRİGGER TOPLAM TUTAR GÜNCELLE AÇIKLAMASI:*

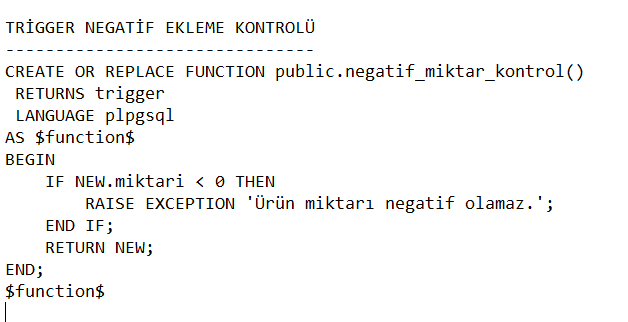
Sepet İd’ si eşleşen sepetin toplam fiyatını miktarı ile satış fiyatını çarparak bu değeri günceller.

*-TRİGGER TOPLAM TUTAR GÜNCELLE ÇALIŞTIRILMASI*



-TRİGGER NEGATİF ÜRÜN EKLEME KONTROLÜ TANIMLANMASI

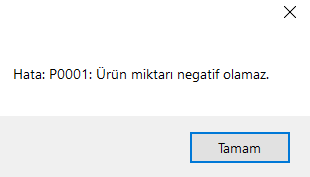
*-TRİGGER NEGATİF ÜRÜN EKLEME KONTROLÜ TANIMLAMASI*

**

*-TRİGGER NEGATİF ÜRÜN EKLEME KONTROLÜ AÇIKLAMASI*

Ürün eklerken eğer negatif miktarda ürün eklenmesi istenirse bu trigger tetiklenir ve buna izin vermez.

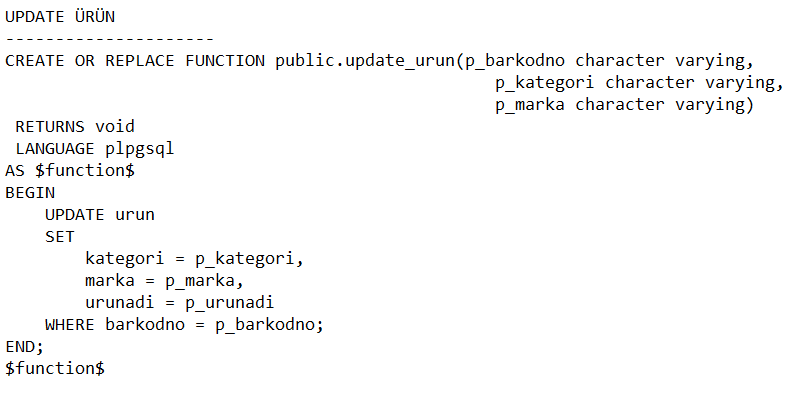
*-TRİGGER NEGATİF ÜRÜN EKLEME KONTROLÜ ÇALIŞTIRILMASI*



FONKSİYONLAR

-UPDATE ÜRÜN FONKSİYONU

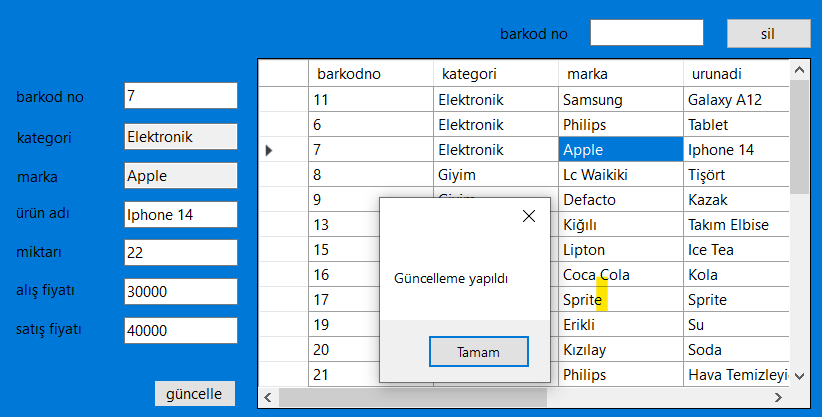
*--UPDATE ÜRÜN FOKNKSİYONU TANIMI :*

**

*-UPDATE ÜRÜN KONKSİYONU AÇIKLAMASI:*

Parametrede verilen barkod no’suna göre ürünün bilgilerini günceller.

*-UPDATE ÜRÜN FONKSİYONU ÇALIŞTIRILMASI:*



1. SONUÇ

Sonuçlar ve Değerlendirme

Projenin Başarıya Ulaşması: Proje, başlangıçta belirlenen hedeflere başarıyla ulaşmıştır. Müşteri ve stok yönetimi, kullanıcı girişleri, ürün ekleme/güncelleme/silme gibi temel işlevsellikler sorunsuz bir şekilde çalışmaktadır. Kullanıcı dostu arayüzü sayesinde müşteriler ve yöneticiler tarafından kolayca kullanılabilmektedir.

Artı Yönler

*Modüler ve Genişletilebilir Tasarım*

Proje, modüler bir yapıya sahiptir, bu da ilerleyen aşamalarda yeni özelliklerin eklenmesini kolaylaştırır. Proje istenilen ölçüde genişletilebilme imkanına sahiptir.

*Güvenli Veri Yönetimi*

PostgreSQL kullanımı ve bu veri tabanına erişimde Npgsql kütüphanesinin kullanılması, güvenli ve tutarlı bir veri yönetimi sağlamıştır. Verilerin gerçek zamanlı olarak programla uyumlu şekilde çalışması sağlanmıştır.

*Performans ve Kullanıcı Dostu Arayüz*

C# ve Windows Forms kullanımı, kullanıcı dostu bir arayüz tasarımına olanak tanımıştır. Bu da kullanıcıların işlemleri hızlı ve etkili bir şekilde gerçekleştirmesini sağlar. Uygulama herkesin kullanabileceği sade bir arayüze sahiptir ve gereksiz fazlalıklar olabildiğince sadeleştirilmeye çalışılmıştır.

Eksi Yönler

*Gelişmiş Raporlama*

Proje şu aşamada detaylı raporlama özelliklerine sahip değildir. İleriki versiyonlarda bu konuda geliştirmeler yapılabilir.

*Web Tabanlı Versiyon*

Şu anki versiyon sadece masaüstü uygulamasıdır. İleride web tabanlı veya mobil sürümü eklenerek daha geniş kitlelere ulaşabilir.

*Mobil Uygulama Desteği*

Proje şu aşamada mobil cihazlar için optimize edilmemiştir. Mobil uygulama desteği eklenerek kullanıcıların mobil cihazlar üzerinden erişim sağlaması hedeflenebilir.

Kazanımlar

*Veri tabanı Tasarımı ve Yönetimi*

PostgreSQL kullanımı sayesinde karmaşık veri yapıları üzerinde etkili çalışma becerisi kazanıldı.

Veri tabanında tablo tasarımı yapma, ilişkiler türetme , stored procedure’ler , trigger’lar ve normalizasyon işlemlerinin yapılmasında deneyim kazanıldı. SQL ve PSQL dilleri kullanılarak bu dillere karşı daha fazla aşinalık kazanıldı.

*Masaüstü Uygulama Geliştirme*

C# ve Windows Forms kullanarak masaüstü uygulama geliştirme konusunda deneyim kazanıldı.

*Ekip Çalışması*

Proje, bir ekip içinde iş birliği yapma yeteneğini geliştirdi. Gerek yüz yüze gerekse uzaktan şekilde iletişim sağlanarak projenin başarıyla tamamlanması için ortaklaşa çalışmalar yapıldı.

*Proje Yönetimi*

Geliştirme aşamalarının planlanması, kodun sürüm kontrolü ve iş takibi konularında deneyim kazanıldı.

İleriki Aşamalar ve Öneriler

*Gelişmiş Özellikler*

Müşterilerin satın alma geçmişine göre hareket edilebilecek eklemeler yapılabilir bu sayede müşteri kendine özgü sayfasıyla daha rahat bir deneyim kazanmış olur.

Ekstra kodlamalar ile oluşabilecek hatalar giderilebilir. Bu sayede programın zaafları giderilmiş olur.

*Raporlama ve Analitik Özellikler*

Detaylı raporlama özellikleri eklenerek yöneticilerin satış verilerini analiz etmeleri kolaylaştırılabilir.

*Web ve Mobil Uyum*

Projeye web tabanlı bir sürüm eklenerek ve mobil uygulama desteği sağlanarak kullanıcı erişimini artırabilir.

*Gelişmiş Güvenlik Özellikleri*

Kullanıcı rolleri ve yetkilendirme sistemi üzerinde çalışarak güvenlik önlemleri arttırılabilir.

*Stok Takibi İyileştirmeleri*

Stok miktarı, minimum stok seviyeleri gibi ek özelliklerle stok takibi daha verimli hale getirilebilir.

*Kullanıcı Geri Bildirimleri*

Kullanıcıların geri bildirimleri alınarak uygulamanın kullanıcı dostu olma özelliği arttırılabilir.

1. KAYNAKÇA

*Youtube*

Projenin hemen hemen her aşamasında bu siteden faydalanılmıştır. C# dilinde , arayüz tasarım aşamalarında , veri tabanı tasarım aşamalarında , diyagramların tasarımlarında , SQL dilinde bilgimizin yeterli olmadığı yerlerde her zaman Youtube kullanılmıştır. Yardımcı olarak en fazla kullandığımız kaynak olmuştur.

*C# ve Windows Forms İçin Resmi Microsoft Dokümantasyonu*

Projenin geliştirilmesinde C# dilinin ve Windows Forms'un resmi Microsoft belgeleri başvuru kaynağı olarak kullanılmıştır.

*PostgreSQL Resmi Dokümantasyonu*

PostgreSQL veritabanının kullanımı ve yönetimi için resmi dokümantasyon kaynak olarak incelenmiştir.

*Npgsql Kütüphanesi Dokümantasyonu*

C# ile PostgreSQL veritabanına erişim sağlamak amacıyla kullanılan Npgsql kütüphanesinin dokümantasyonu kaynak olarak kullanılmıştır.

*W3Schools ve Stack Overflow*

C# diline ve Windows Forms kullanımına dair karşılaşılan sorunların çözümü için W3Schools ve Stack Overflow gibi çevrim içi kaynaklar kullanılmıştır.

*Forumlar*

Yaşanan bir sürü problem için internetteki onlarca forumda yer alan tartışmalardan faydalanılırmıştır.