PROJE RAPORU



**PROJE**

**RAPOR TESLİM TARİHİ PROJE ADI HAZIRLAYANLAR:**

|  |  |  |
| --- | --- | --- |
| **15.12.2023** | **Dominos Sistemi** | **Ahmet Akif KASIM**  **Betül DAŞÇI**  **Merve KAVLAK**  **Sena Ceren KURNAZ**  **Zeynep Azranur YÜKSEL** |

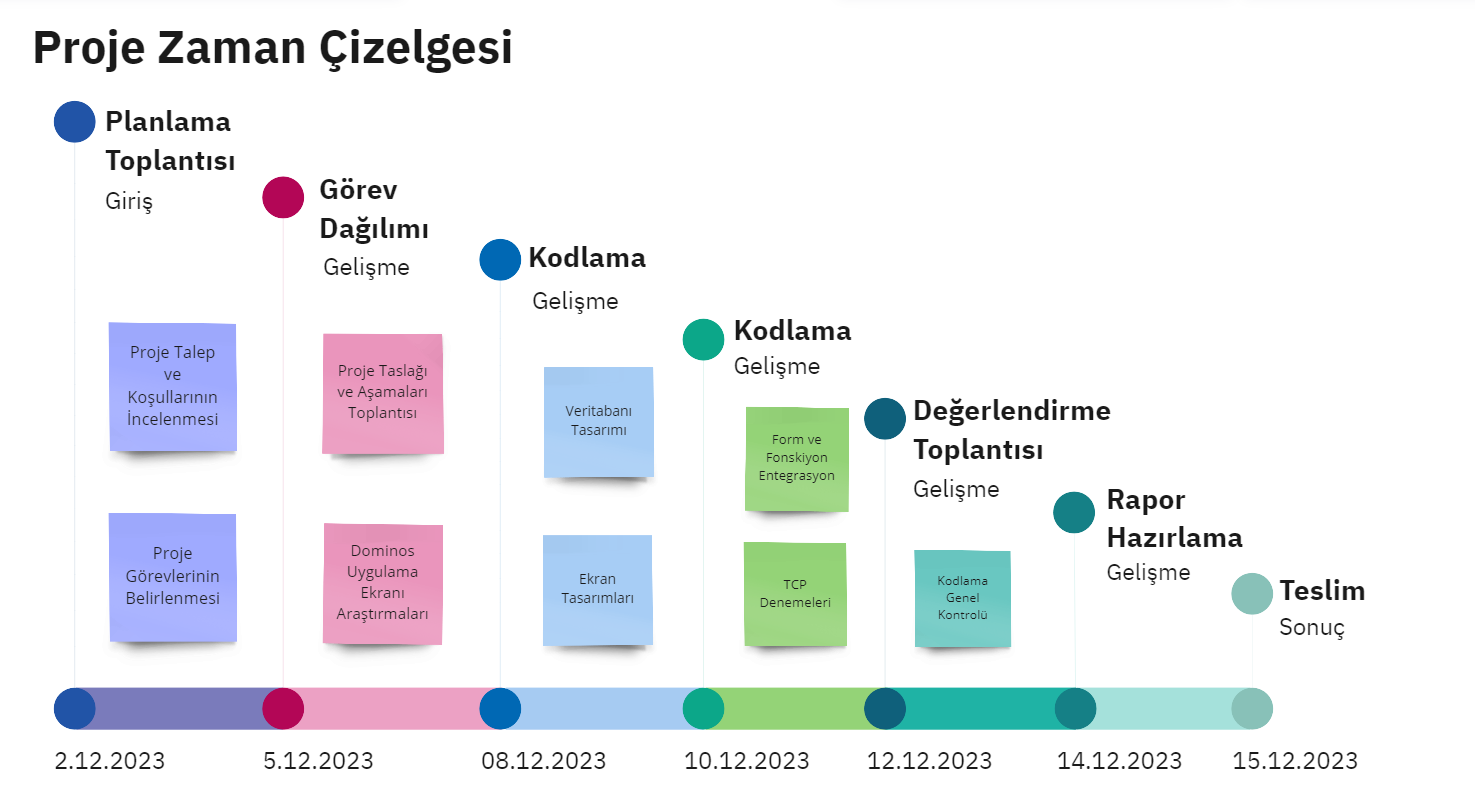


**PROJE ÖZETİ**

**Hazırlanmış olan Dominos Ekran Yönetim Sistemi; genel alım satım faaliyetlerini barındıran ekran yönetim sistemi olarak tasarlanmıştır. Hazırlanan sistem tasarımı login işlemleri, ürün satışı ve alışı, ürün ve sipariş listeleme, sipariş onayı, toplam tutar bilgisi gibi işlemleri barındırmaktadır. Ahmet Akif KASIM, Betül DAŞÇI, Merve KAVLAK, Sena Ceren KURNAZ, Zeynep Azranur YÜKSEL tarafından 1 Aralık – 15 Aralık arasında geliştirilmiştir.**



|  |  |
| --- | --- |
| **GÖREV** | **YAPILAN İŞLEMLER** |
| **Proje Taslaklarının Hazırlanması** | Miro web sitesi kullanılarak hazırlanan taslaklar ile proje tanımı somutlaştırıldı. |
| **Temel Tasarımlı Form Ekranlarının Hazırlanması** | Backend sürecinde, hem çalışma verimi ve hızını arttırabilmek hem de hata payını azaltmak adına geçici olarak bir primitif tasarım oluşturuldu. Çekilen fotoğraflardaki tasarıma göre ekran tasarımları oluşturuldu. |
| **Dominos Uygulama Ekranı Araştırması** | Fiziksel olarak mekanlara gidilerek fotoğraf çekmeye çalışıldı. Erişilemeyen kısımlar için sosyal medya üzerinden dominos çalışanlarının reelslarından hareketle veri elde edilmeye çalışıldı. |
| **Ürünler Dizisinin Oluşturulması** | Ürün isimleri, ek materyal bilgileri, ürün fiyat bilgileri tanımlandı. |
| **Ürünler Fonksiyonlarının Oluşturulması** | Ürün ekleme ve listeleme fonksiyonları yazıldı. |
| **Ürün Satış Bölümü İşlevlerinin Oluşturulması** | * Ürün bilgileri ilgili sayfada butonlar üzerinde listelendi. * Butonlara tıklandığında sipariş bilgisi alındı. * Toplam Tutar belirtildi. * Ürünün satılması ile Mutfak ekranı arasındaki entegrasyon kuruldu. |
| **Mutfak Kanadından Sipariş Onay ve Diğer Hizmetler** | * Sipariş alış formu oluşturularak ürünlerden kaç adet alındığı ve ücret bilgileri belirtildi. * Sipariş hazırlanma süresi ve siparişin hazırlandığını belirten bir metot oluşturulfu. |
| **Admin Girişinin Eklenmesi** | Yalnızca market personellerinin erişim izninin olduğu login ekranı oluşturuldu. |
| **Proje Raporu Hazırlama** | Proje özeti, proje süreci ve süreç içerisinde karşılaşılan sorunlar ve çözümlerine yer verilen proje raporu hazırlandı. |

****

**PROJE TASLAĞI**

**metin, ekran görüntüsü, Post-it notu, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu**

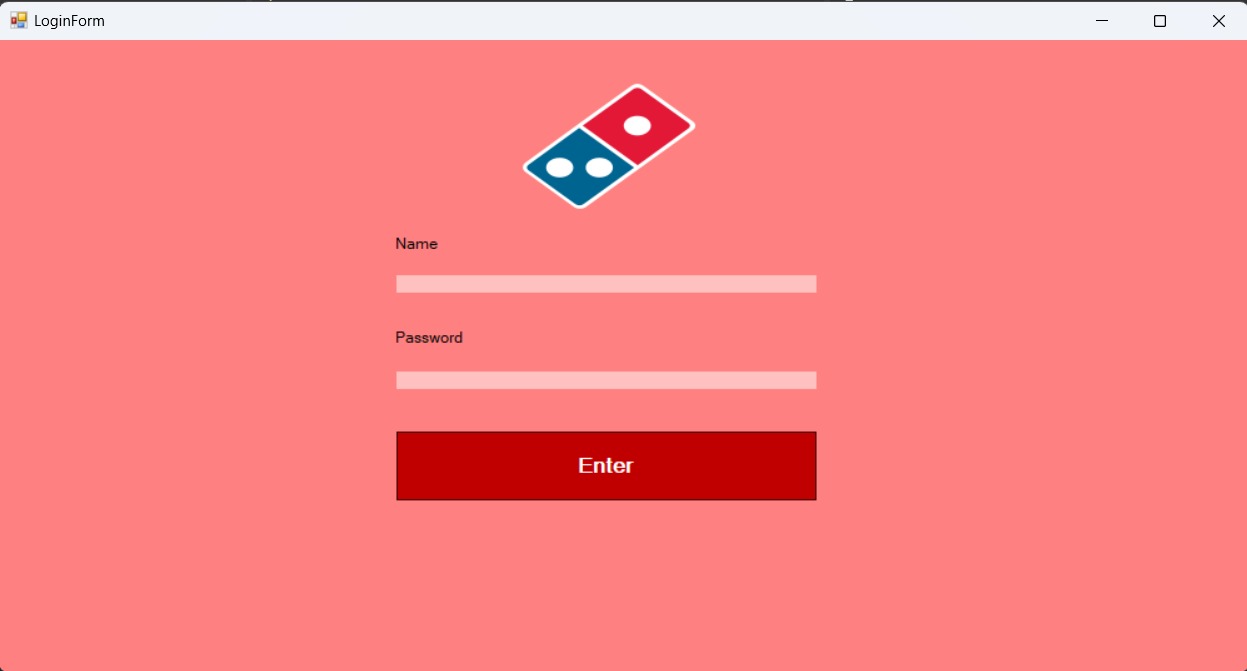
Bu proje yol haritasında, Dominos şubelerindeki uygulama ekranlarını detaylı bir şekilde inceleyerek bir taslak tasarım oluşturduk. Bu süreçte 'Dominos Ekran Yönetim Sitesi' isimli bir çatı altında, çeşitli modüller geliştirdik.

Öncelikle, personelin sistemle etkileşimini başlatan bir 'Login' (Giriş) ekranı tasarladık. Ardından, mutfağın siparişleri etkin bir şekilde görebilmesi için 'Mutfak Ekranı' oluşturduk; bu ekran üzerinden sipariş görüntüleme, hazırlama süreci, onayı ve TCP iletişimi gibi özellikleri entegre etmeye çalıştık. 'Order Ekranı' (Sipariş Ekranı) modülünde, ürün çeşitleri, boyutları ve sipariş özeti gibi özelliklere yer verdik, böylece sipariş alma ve onay süreçlerini tamamlamış olduk. Müşteri ekranında ise sipariş numarası, bekleme süresi ve hazırlandı bilgisi gibi müşteriler için haber niteliği taşıyan bilgileri bir araya getirdik.

Genel hatlarıyla kullanıcı deneyimini iyileştirmeyi amaçlayan bu bileşenleri, işlevselliği ön planda tutarak yeniden oluşturmayı denedik. Bu yol haritası Dominos şubelerindeki operasyonel işleyişi yansıtmayı hedeflemektedir.

**Proje Taslağına Dair Açıklamalarıyla Örnekler**

Login Ekranımız:



Aşağıdaki kod parçaları, iki farklı form arasındaki etkileşimi yöneten ve kullanıcı tarafından tetiklenen olaylara yanıt olarak arayüzde güncellemeler gerçekleştiren iki metodu gösterir. İlk metot, enterBtn\_Click, bir kullanıcının kimlik doğrulamasını yönetir; ikinci metot, personelLabel, bilgi gösterimi için bir kullanıcı arayüz bileşenini günceller. enterBtn\_Click metodu, LoginForm sınıfının bir parçası olarak tanımlanmıştır ve bir giriş butonuna tıklama olayına yanıt vermek üzere tasarlanmıştır.

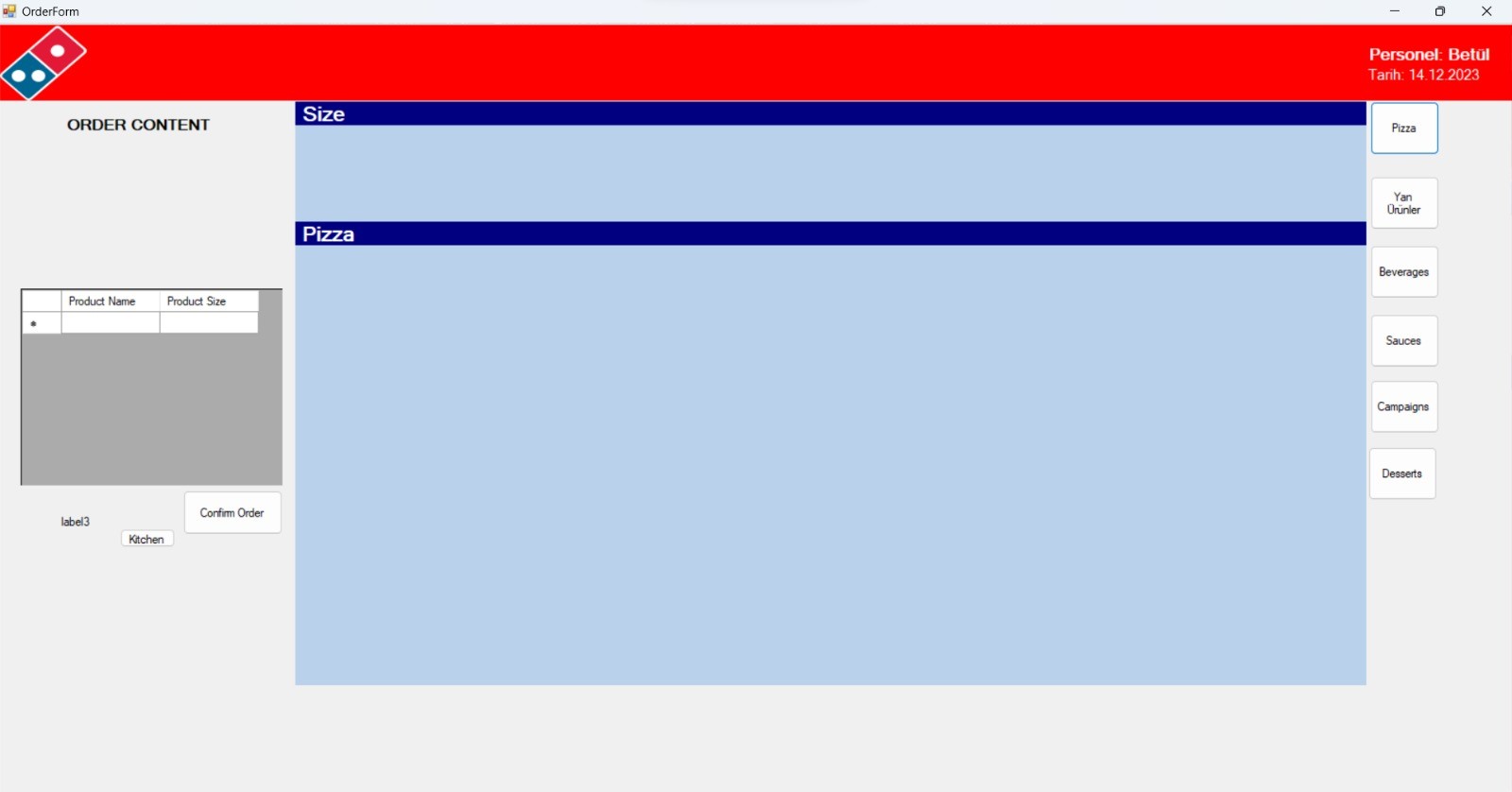
metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu metot, kullanıcı tarafından girilen adı (nameTxt.Text) kullanarak veritabanında (db) bir Personel nesnesi arar. Bu arama, LINQ sorgusu kullanılarak yapılır ve eşleşen ilk nesneyi veya hiçbir şey bulunamazsa null değerini döndürür.

Eğer bir Personel nesnesi bulunursa, metot, ilgili personel bilgileri ile OrderForm nesnesini başlatır ve bu form üzerindeki personel adı etiketini (PersonelLbl) güncellemek için personelLabel metodunu çağırır. Ardından OrderForm ekranını modal olarak gösterir. personelLabel metodu, OrderForm sınıfında tanımlıdır ve aldığı name parametresi ile PersonelLbl etiketindeki metni günceller. Bu etiket, arayüzdeki personel bilgisini gösteren bir etikettir. Metot, bu etikete dinamik olarak personelin adını ekler, böylece kullanıcıya hangi personelin oturum açtığını bilgi olarak sunar.

Özet olarak bu kod blokları, yazılım geliştirme sürecinde kullanıcıların etkileşimlerine dayalı olarak sınıflar arası bilgi aktarımını ve arayüz güncellemelerini etkin bir şekilde gerçekleştiren yöntemleri örnekler. Ayrıca, bu yaklaşım, nesne yönelimli programlama paradigmasının temel prensiplerinden biri olan veri kapsüllemesini ve etkileşimli kullanıcı arayüzlerinin dinamik olarak güncellenmesini gösterir.

Login işlemi gerçekleştirildikten sonra karşımıza çıkan Order Ekranı:  
  


Ekrandaki müşteri siparişine göre basılacak olan butonlar, aşağıdaki fonksiyonu aktifleştirerek ekranda Size ve Pizza panelleriyle ilgili optionları yansıtmayı sağlar.



Bu fonksiyon, kullanıcı arayüzünde dinamik olarak butonlar oluşturan ve bu butonlara etkileşim özelliği ekleyen bir C# metodur. ButtonCreate metodu, bir Panel nesnesi, bir ürünün başlığı ve fiyatı alarak çalışır. Metodun içerisinde, belirli bir düzen içinde butonlar oluşturulur ve her bir buton, üzerinde ürünün adı ve fiyat bilgisi gösterilecek şekilde ayarlanır.

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

Açıklama otomatik olarak oluşturuldu

Eğer xStart değişkeni 800'den küçükse, yani butonlar için yatayda yer varsa, buton belirli bir konumda (xStart, yStart) oluşturulur ve panel içine eklenir. Her buton oluşturulduğunda, xStart koordinatı arttırılarak yeni buton için yer açılır.

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu bilgiler kullanılarak, productCrud nesnesi üzerinden bir ürün sorgulaması yapılır. Eğer ürün bulunursa, global products listesine eklenir ve toplam fiyat güncellenir. Son olarak, sipariş bilgileri bir DataGridView kontrolüne eklenir ve toplam fiyat etiketi (TotalPriceLbl) güncellenir.

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Eğer xStart değeri 800'den büyükse, bu durum yatayda yer kalmadığını gösterir ve butonlar yeni bir satırda oluşturulmaya başlar. Bu durumda yStart dikey konumu arttırılır ve xStart yeniden başlangıç konumuna ayarlanır. Ardından yeni buton oluşturulup işlemler tekrarlanır.

Fonksiyonumuz, kullanıcı arayüzünde ürün seçimi ve sipariş yönetimi işlevlerini kolaylaştırmak için tasarlanmış etkileşimli bir yapı sunar. Bu yapı sayesinde, kullanıcıların ürün seçimini ve sipariş toplamını gerçek zamanlı olarak görmeleri ve yönetmeleri sağlanır.

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Yukarıdaki fonksiyonu, bir sipariş kaydetme işlemini yöneten ve siparişin veritabanına başarıyla eklenmesinin ardından kullanıcı arayüzünü sıfırlayan bir C# metodu olarak oluşturduk. ConfirmBtn\_Click metodu, belirli bir olay tetiklendiğinde çalıştırılarak: Metot içerisinde, öncelikle bir sipariş listesinin varlığını kontrol eder; eğer ürün listesi boş değilse işleme devam eder.

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Metodun ana işlevi, bir Order nesnesi oluşturarak bu nesneye mevcut zamanı, toplam fiyatı, personel kimliğini ve durum bilgilerini atamaktır. Daha sonra da bu nesneyi, db içindeki Order koleksiyonuna ekler. SaveChanges metodunu çağırarak yapılan değişiklikleri veritabanına kaydeder. Bir DbUpdateException hatası oluşursa, bu durumu kullanıcıya bir mesaj kutusu aracılığıyla bildirir ve işlemi durdurur.

Her bir ürün için OrderProduct nesneleri oluşturulur ve her ürünün kategorisine göre boyutu ve miktarı atanır. Bu OrderProduct nesneleri, bir listeye eklenir ve bu liste db içindeki OrderProduct koleksiyonuna topluca eklenir. İkinci bir SaveChanges çağrısıyla, bu ilişkisel veriler de veritabanına kaydedilir. Bu noktada bir hata oluşursa, kullanıcıya hata mesajı gösterilir ve işlem yine durdurulur.

Sipariş işlemleri başarıyla tamamlandığında, sipariş ürün listesi ve arayüzdeki sipariş bilgileri temizlenir. Toplam fiyat sıfırlanır ve kullanıcıya siparişin başarıyla alındığına dair bir onay mesajı gösterilir. Eğer ürün listesi başlangıçta boş ise, kullanıcıya sipariş içeriğinin boş olduğuna dair bir uyarı mesajı gösterilir.

Bu metod, kullanıcı eylemlerine dayalı olarak veritabanı işlemlerini yürüten ve arayüzle ilgili güncellemeleri gerçekleştiren bir işlevsel mekanizma olarak görülebilir. Aynı zamanda, veritabanı ile etkileşimde bulunurken olası hataları ele almak için gerekli kontrolleri de içerir.

Müşteri bekleme ekranımızda yukarıdakilere ek olarak mutfakta onayı alınan siparişin yapılma süreleri ve isimleri ekrana yazdırılır. Ve tamamlandıktan sonra da yeşil alanda sipariş ismiyle gösterilir.

metin, ekran görüntüsü, dikdörtgen, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Kodlama Sürecinde Karşılaşılan Sorunlar ve Çözümleri**

Hata 1: **Database Hazırlama Sürecinde Product Size Ekleme Kararsızlığı**

Veritabanı hazırlık sürecinde, ürün boyutlarını nasıl ele alacağımız konusunda önemli bir tercih yapmamız gerekti. Başlangıçta, boyutları ürünler tablosuna bir özellik olarak dahil etme fikrini düşündük. Ancak, yapılan araştırmalar ve değerlendirmeler sonucunda, boyutların ayrı bir tablo olarak eklenmesinin daha uygun olacağı sonucuna vardık. Bu karar, veritabanı normalizasyonu ilkelerine uygun bir şekilde, veri bütünlüğünü ve esnekliğini artırmayı amaçlamaktadır

Hata 2: **SQL’de Atanan Tablo Verilerini Gösterme Hatası**

Kod yazım aşamasında ise, ürün boyutlarını bu yeni oluşturulan tablodan entegre ettik. Bu sayede, ürünlerin boyut bilgilerini daha düzenli ve sistemli bir biçimde yönetebildik.

Sipariş işlemleri kapsamında, siparişlerin Orders ve OrderProducts tablolarına doğru şekilde eklenmesi sürecini ele aldık. İlk denemelerimizde, SQL veritabanında oluşturduğumuz siparişleri doğru bir şekilde görüntüleyememe sorunu ile karşılaştık. Bu problemi detaylıca inceledikten sonra, seçilen ürünleri buton üzerinde görünen isimle doğrudan seçmemizden kaynaklandığını fark ettik.

Sorunu çözmek için, her butonun click olayında, seçilen ürünün özelliklerini OrderForm içerisinde oluşturduğumuz Products listesine ekledik. Bu işlem sırasında, selectedProduct adında bir değişken tanımladık ve bu değişkeni tüm süreçlerde kullanarak tutarlılık sağladık. Bu metodoloji, veritabanı işlemlerinde karşılaştığımız sorunların üstesinden gelmemize ve veri akışını daha verimli bir şekilde yönetmemize olanak tanıdı.

**Sonuç**

Bu rapor, projenin geliştirilme sürecinde karşılaşılan bir sorunu nasıl aştığımızı göstermektedir. Sorunun tespiti ve çözümü, proje ekibinin iş birliğiyle başarılı bir şekilde gerçekleştirildi. Bu deneyim, gelecekteki projelerde benzer sorunların daha hızlı ve etkili bir şekilde çözülmesine yardımcı olacaktır.