**T.C.**

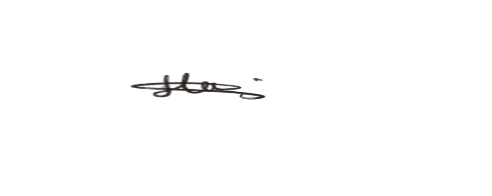
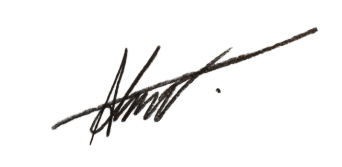
**BANDIRMA ONYEDİ EYLÜL ÜNİVERSİTESİ**

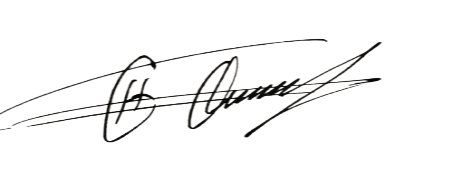
**MÜHENDİSLİK VE DOĞA BİLİMLERİ ENSTİTÜSÜ**



**COMPANY MANAGMENT SYSTEM RAPOR**



**Hazırlayanlar  
Emre Kaya  
Hale Nur Sarışen**

**Ahmet Fevzi Can**

**Göktuğ Altıntaş**

**İÇİNDEKİLER**

* **İşe Alım Özeti**
* **Mail Gönder**
* **Mail Oku Özeti**
* **Eleman Yönetim Özeti**
* **Şifremi Unuttum Özeti**
* **İş Listesi Özeti**
* **Dökümantasyon Özeti**
* **Giriş Ekranı ve Çıkış Özeti**
* **Anasayfa Ekranı Özeti**
* **Butonlar Özeti**
* **UI/UX Özeti**
* **Grup Tabloları ve Grafikleri**

**IseAlim Sınıfı**

**Veritabanı Bağlantısı ve Komut Tanımlama**

* **Veritabanı Bağlantısı ve Komut Nesnesi Oluşturma:**
  + SqlConnection ve SqlCommand nesneleri oluşturularak veritabanına bağlantı sağlanır.

**ComboBox Veri Kaynakları**

* **ComboBox Veri Kaynakları:**
  + İlgili ComboBox'lar için önceden tanımlanmış veri listeleri atanır.

**Departman ve Yetki Seviyesi İlişkilendirmesi**

* **Departman ve Yetki Seviyesi İlişkilendirmesi:**
  + ComboBox'ların seçim değiştirdiğinde ilgili olaylar tetiklenerek departman seçimine bağlı olarak yetki seviyesi seçenekleri güncellenir.

**Kullanıcı Bilgilerini Kaydetme**

* **Kullanıcı Bilgilerini Kaydetme:**
  + "Kaydet" butonuna tıklandığında, kullanıcının girdiği bilgileri kullanarak SQL sorgusu oluşturulup veritabanına eklenir.

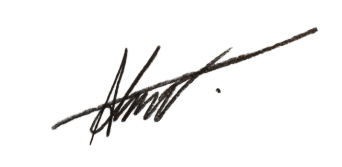
**Hata Kontrolü ve Geri Dönüş**

* **Hata Kontrolü ve Geri Dönüş:**
  + Veritabanı işlemleri sırasında oluşabilecek hatalar yakalanır ve kullanıcıya bilgi mesajları gösterilir.

**Geri Dönme İşlemi**

* **Geri Dönme İşlemi:**
  + "Geri" butonuna tıklandığında, pencere gizlenir.



****

**MailGönder Sınıfı**

**Constructor (Yapıcı Metod)**

* **Constructor Oluşturma:**
  + Kullanıcı adını alan bir constructor metodu oluşturulur. Kullanıcı adı TextBox'a atanır ve kullanıcının değiştirmesine izin verilmez.

**Gönderme İşlemi**

* **Gönderme İşlemi:**
  + "Gönder" butonuna tıklandığında, kullanıcının girdiği bilgilerle birlikte MailGönderici metodu çağrılır.
  + Bilgiler veritabanına eklenir ve başarılı bir şekilde gönderildiğine dair bir bildirim penceresi görüntülenir.
  + Gönderildikten sonra giriş kutuları temizlenir ve Departman ComboBox'ı sıfırlanır.

**Mail Gönderici Metodu**

* **Mail Gönderici Metodu:**
  + Veritabanına mail bilgilerini eklemek için kullanılır.
  + SqlConnection ve SqlCommand nesneleri kullanılarak veritabanına bağlantı sağlanır.
  + Parametreler eklenir ve SQL sorgusu çalıştırılır.

**Geri Dönme İşlemi**

* **Geri Dönme İşlemi:**
  + "Geri" butonuna tıklandığında, pencere gizlenir.



**MailleriOku Sınıfı**

**Constructor (Yapıcı Metod)**

* **Constructor Oluşturma:**
  + Kullanıcı adı, yetki seviyesi ve departman bilgilerini alan bir constructor metodu oluşturulur.

**Departman Bilgisi Alımı**

* **Departman Bilgisi Alımı:**
  + Veritabanından belirli bir kullanıcı adına göre departman bilgisini çeken bir metot bulunur.

**Kontrolleri Sıfırlama**

* **Kontrolleri Sıfırlama:**
  + ComboBox ve ListView nesnelerini temizleyen bir metot bulunur.

**Kullanıcı Adını Label'a Eklemek**

* **Kullanıcı Adını Label'a Eklemek:**
  + Form yüklenirken kullanıcı adını ekrana ekleyen bir metot bulunur.

**Departman Ekleme**

* **Departman Ekleme:**
  + ComboBox'a departman ekleyen metotlar bulunur.

**Kullanıcıya Özel İşlemler**

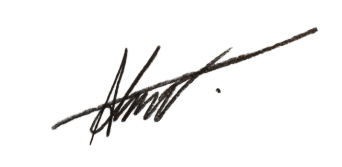
* **Kullanıcıya Özel İşlemler:**
  + Kullanıcının yetki seviyesine ve departmanına göre belirli işlemler gerçekleştiren bir switch-case yapısı bulunur.

**Departman ve Alıcı Bilgilerini ComboBox'a Ekleme**

* **Departman ve Alıcı Bilgilerini ComboBox'a Ekleme:**
  + ComboBox'lar seçildiğinde belirli departman ve alıcı bilgilerini ekleyen metotlar bulunur.

**Alıcı Seçildiğinde Mailleri Gösterme**

* **Alıcı Seçildiğinde Mailleri Gösterme:**
  + ComboBox'ta alıcı seçildiğinde, ilgili alıcıya ait mailleri veritabanından çeken ve ListView'e ekleyen bir metot bulunur.

****

**ElemanYönetimi Sınıfı**

**Constructor (Yapıcı Metod)**

* **Constructor Oluşturma:**
  + Veritabanından elemanları listelemek için bir constructor metodu oluşturulur.
  + ComboBox'ları doldurmak için önceden tanımlanan listeler kullanılır.

**Departman ve Yetki Seviyesi Seçimine Göre ComboBox Güncelleme**

* **ComboBox Güncelleme:**
  + Departman seçimi değiştiğinde, buna bağlı olarak departman seviye ComboBox'ını güncelleyen bir metot bulunur.
  + Yetki seviyesi seçimi değiştiğinde, buna bağlı olarak yetki seviyesi ComboBox'ını güncelleyen bir metot bulunur.

**Elemanları Listeleme**

* **Elemanları Listeleme:**
  + Veritabanından eleman bilgilerini çekerek ListView'e ekleyen bir metot bulunur.

**Eleman Güncelleme**

* **Eleman Güncelleme:**
  + Seçilen elemanın bilgilerini güncelleyen bir metot bulunur.
  + Gerekli alanları kontrol eder ve güncelleme işlemi gerçekleştirir.

**Eleman Bilgilerini Doldurma**

* **Eleman Bilgilerini Doldurma:**
  + ListView'den seçilen elemanın bilgilerini ilgili TextBox'lara dolduran bir metot bulunur.

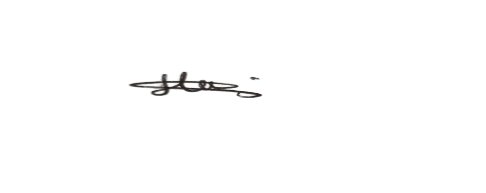
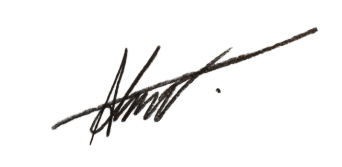
**Eleman Silme**

* **Eleman Silme:**
  + ListView'den seçilen elemanı veritabanından silen bir metot bulunur.

**Formu Kapatma**

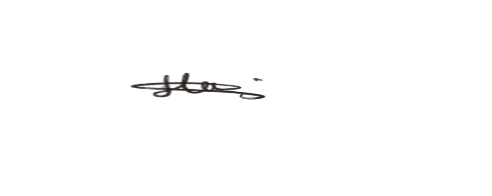
* **Formu Kapatma:**
  + Formu kapatma işlemini gerçekleştiren bir metot bulunur.

**PDF Oluşturma**

* **PDF Oluşturma:**
  + Veritabanındaki eleman bilgilerini PDF dosyasına aktaran bir metot bulunur.
  + ****iTextSharp kütüphanesi kullanılarak PDF oluşturulur.

**Şifre Sıfırlama Formu Özet:**

1. **Bağlantı Kurma:**
   * Veritabanına bağlantı sağlanır.
   * Bağlantı bilgileri **connectionString** özelliği ile belirlenir.
2. **Form Başlangıcı:**
   * **SqlConnection** sınıfı kullanılarak bağlantı oluşturulur.
   * Gerekli ayarlamalar yapılır.
3. **Şifre Sıfırlama İşlemi:**
   * **ResetPassword** metodu, kullanıcının şifresini günceller.
   * Gizli soru ve cevap doğrulaması yapılır.
4. **Şifre Sıfırlama Butonu:**
   * Kullanıcıdan alınan bilgilerle şifre sıfırlama işlemi gerçekleştirilir.
   * Hata durumunda kullanıcıya bilgi mesajları gösterilir.
5. **Geri Dönme Butonu:**
   * Form kapatılarak ana ekrana dönülür.

****

**İş Listesi Sınıfı**

**Constructor (Yapıcı Metod)**

* **Constructor Oluşturma:**
  + Formun başlatılması sırasında kullanıcı adı, yetki seviyesi ve departman bilgilerini alır.
  + Gerekli kontrolleri gerçekleştirerek yetki seviyesine göre arayüzü günceller.

**Departman ve Yetki Seviyesine Göre Arayüz Güncelleme**

* **Arayüz Güncelleme:**
  + Departman ve yetki seviyesine göre arayüzdeki bazı öğelerin görünürlüğünü ve etkinliğini düzenleyen metotlar bulunur.

**İş Listelerini Doldurma**

* **İş Listelerini Doldurma:**
  + Seçilen departmana göre yapılacak işler, devam eden işler ve biten işlerin listelerini dolduran bir metot bulunur.
  + Bu metot, departman değiştikçe çağrılır.

**İş Ekleme**

* **İş Ekleme:**
  + Yetki seviyesine bağlı olarak iş eklemek için kullanılan bir metot bulunur.
  + İş ekledikten sonra ilgili listeleri günceller ve veritabanına kaydeder.

**İş Başlatma ve Bitirme**

* **İş Başlatma ve Bitirme:**
  + Yapılacak işler listesinden işi devam eden işler listesine taşımak veya devam eden işler listesinden biten işler listesine taşımak için kullanılan metotlar bulunur.
  + İş durumu değiştiğinde veritabanına kayıt işlemi gerçekleştirilir.

**Verileri Veritabanına Kaydetme**

* **Veritabanına Kaydetme:**
  + Çalışma sayısı tablosuna kullanıcı adı ve yapilan iş bilgilerini kaydeden metotlar bulunur.
  + İş başlatma ve bitirme işlemlerinde kullanılır.

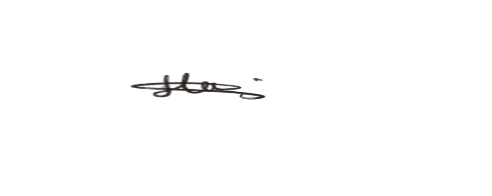
**Mail Gönderici**

* **Mail Gönderici:**
  + İş eklediğinizde çağrılan bir metot bulunur.
  + İş eklenen departmana mail gönderir.

**PDF Oluşturma**

* **PDF Oluşturma:**
  + Veritabanındaki çalışma sayısı bilgilerini PDF dosyasına aktaran bir metot bulunur.
  + iTextSharp kütüphanesi kullanılarak PDF oluşturulur.

**Geri Dönme ve Form Kapatma**

* ** Geri Dönme ve Form Kapatma:**
  + Formu gizleme işlemlerini gerçekleştiren metotlar bulunur.

**MailleriDökümante Sınıfı**

**Constructor (Yapıcı Metod)**

* **Constructor Oluşturma:**
  + Form başlatılırken, arayüzün hazırlanması ve departman bilgilerini içeren ComboBox'ın doldurulması işlemleri gerçekleştirilir.

**Mailleri Listeleme**

* **Mailleri Listeleme:**
  + Veritabanından mail bilgilerini çekerek ListView kontrolüne ekleyen bir metot bulunur.
  + Gönderilen mail, kullanıcı adı, alıcı, departman ve başlık bilgilerini içerir.

**PDF Oluşturma**

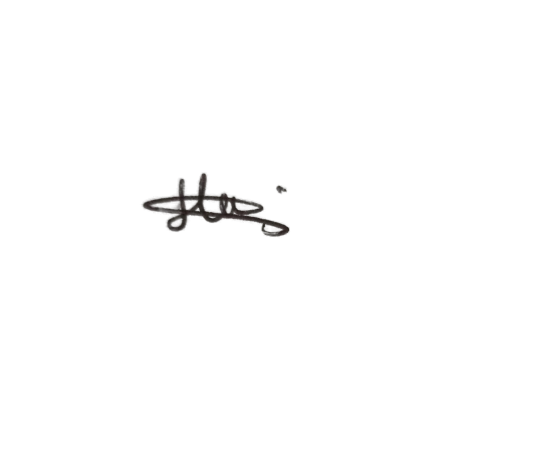
* **PDF Oluşturma:**
  + Seçilen departmandaki mailleri PDF olarak kaydetme işlevini gerçekleştiren bir metot bulunur.
  + iTextSharp kütüphanesi kullanılarak PDF oluşturulur.

**Veritabanından Mailleri Seçilen Departmana Göre PDF'e Aktarma**

* **Veritabanından Mailleri Seçilen Departmana Göre PDF'e Aktarma:**
  + Seçilen departmandaki mailleri filtreleyerek PDF'e dönüştürme işlemlerini gerçekleştirir.
  + dosyasının adını ve kayıt konumunu kullanıcıya belirletir.

**Geri Dönme ve Form Kapatma**

* **Geri Dönme ve Form Kapatma:**
  + Formu gizleme işlemlerini gerçekleştiren bir metot bulunur.

****

**Login Sınıfı**

**Constructor (Yapıcı Metod)**

* **Constructor Oluşturma:**
  + Form başlatıldığında, kullanıcı adını daha önce hatırlayan bir dosya varsa bu bilgiyi alan bir constructor bulunur.
  + Hatırlama seçeneği seçili ise, şifre alanı temizlenmez.

**Kullanıcı Girişi Doğrulama**

* **Kullanıcı Girişi Doğrulama:**
  + Kullanıcı adı ve şifre bilgilerini alarak veritabanında doğrulama işlemlerini gerçekleştiren bir metot bulunur.
  + Kullanıcı adı ve şifre doğru ise **true**, değilse **false** döner.

**Kullanıcı Bilgilerini Dosyaya Kaydetme**

* **Kullanıcı Bilgilerini Dosyaya Kaydetme:**
  + Kullanıcı adını bir dosyaya kaydetme işlemlerini gerçekleştiren bir metot bulunur.

**Kullanıcı Bilgilerini Dosyadan Okuma**

* **Kullanıcı Bilgilerini Dosyadan Okuma:**
  + Daha önce kaydedilmiş olan kullanıcı adını dosyadan okuma işlemlerini gerçekleştiren bir metot bulunur.

**Kullanıcı Bilgilerini Dosyadan Silme**

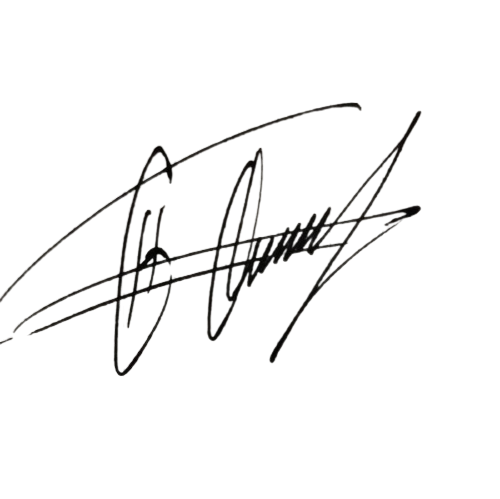
* **Kullanıcı Bilgilerini Dosyadan Silme:**
  + Kullanıcı adının kaydedildiği dosyayı silme işlemlerini gerçekleştiren bir metot bulunur.

**Kullanıcı Departman ve Yetki Bilgilerini Sorgulama**

* **Kullanıcı Departman ve Yetki Bilgilerini Sorgulama:**
  + Veritabanından kullanıcının departman ve yetki bilgilerini sorgulama işlemlerini gerçekleştiren metotlar bulunur.

**Şifremi Unuttum Formunu Açma**

* **Şifremi Unuttum Formunu Açma:**
  + "Şifremi Unuttum" butonuna tıklanması durumunda bu butona bağlı formun açılmasını sağlayan bir metot bulunur.

****

**Anasayfa Sınıfı**

**Constructor (Yapıcı Metod)**

* **Constructor Oluşturma:**
  + İkinci ekran başlatıldığında, kullanıcı adı, yetki seviyesi ve departman bilgilerini alan bir constructor bulunur.
  + Kullanıcının yetkisine göre butonların görünürlüğü ve etkinliği güncellenir.

**Yetkiye Göre Buton Güncelleme**

* **Yetkiye Göre Buton Güncelleme:**
  + Kullanıcının yetki seviyesine göre bazı butonların görünürlüğünü ve etkinliğini kontrol eden bir metot bulunur.
  + Yetki seviyesi 1 olan (Yönetici) kullanıcı, işe alım, yönetim, mail dokümantasyonu gibi butonlara erişebilir.
  + Yetki seviyesi 2 olan (Çalışan) kullanıcı, sadece iş listesi ve mail yazma butonlarına erişebilir.
  + Yetki seviyesi 3 olan (Stajyer) kullanıcı, hiçbir özel yetkiye sahip olmadığı için tüm butonlara erişemez.
  + Diğer durumlar için varsayılan bir durum bulunur.

**İşe Alım Formunu Açma**

* **İşe Alım Formunu Açma:**
  + "İşe Alım" butonuna tıklanması durumunda, işe alım formunu açan bir metot bulunur.

**Eleman Yönetimi Formunu Açma**

* **Eleman Yönetimi Formunu Açma:**
  + "Eleman Yönetimi" butonuna tıklanması durumunda, eleman yönetimi formunu açan bir metot bulunur.

**İş Listesi Formunu Açma**

* **İş Listesi Formunu Açma:**
  + "İş Listesi" butonuna tıklanması durumunda, iş listesi formunu açan bir metot bulunur.

**Mail Gönderme Formunu Açma**

* **Mail Gönderme Formunu Açma:**
  + "Mail Yaz" butonuna tıklanması durumunda, mail gönderme formunu açan bir metot bulunur.

**Çıkış Yapma**

* **Çıkış Yapma:**
  + "Çıkış" butonuna tıklanması durumunda, kullanıcıyı çıkış ekranına yönlendiren bir metot bulunur.

**Mail Dokümantasyon Formunu Açma**

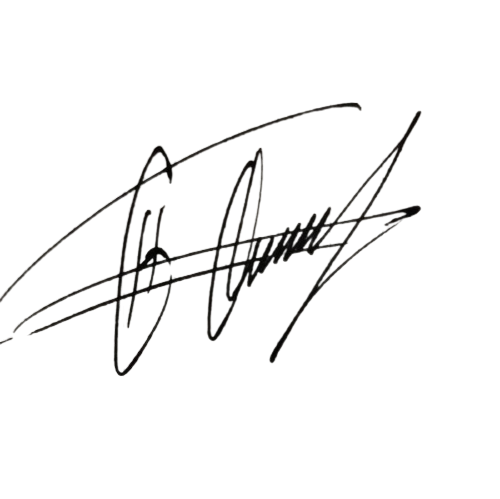
* **Mail Dokümantasyon Formunu Açma:**
  + "Mail Dokümantasyonu" butonuna tıklanması durumunda, mail dokümantasyon formunu açan bir metot bulunur.

**Bilgilendirme Formunu Açma**

* **Bilgilendirme Formunu Açma:**
  + "Bilgilendirme" butonuna tıklanması durumunda, bilgilendirme formunu açan bir metot bulunur.

**Mail Okuma Formunu Açma**

* **Mail Okuma Formunu Açma:**
  + "Mail Oku" butonuna tıklanması durumunda, mail okuma formunu açan bir metot bulunur.

****

**Özel Buton Tasarımı**

İlk olarak, kullanıcı tarafından belirlenebilen özel buton tasarımı oluşturmak amacıyla **CustomBtn** adında bir sınıf oluşturduk. Bu sınıf, butonların kenarlık kalınlığı, yuvarlaklık derecesi, kenarlık rengi gibi özellikleri belirlemek için tasarlanmıştır.

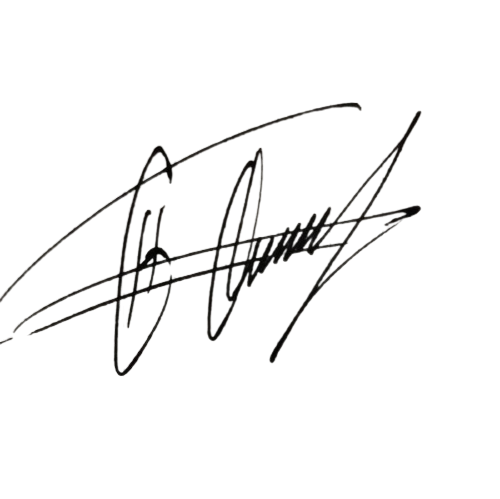
**CustomBtn Sınıfı Özellikleri**

1. **BorderSize (Kenarlık Kalınlığı):**
   * Butonun kenarlık kalınlığını belirler.
   * Kullanıcı, tasarım sırasında bu özelliği ayarlayabilir.
   * Örnek Kullanım: **customBtn1.BorderSize = 2;**
2. **BorderRadius (Kenar Yarıçapı):**
   * Butonun kenar yarıçapını belirler.
   * Kullanıcı, tasarım sırasında butonun ne kadar yuvarlak olacağını ayarlayabilir.
   * Örnek Kullanım: **customBtn1.BorderRadius = 10;**
3. **BorderCoIor (Kenarlık Rengi):**
   * Butonun kenarlık rengini belirler.
   * Kullanıcı, tasarım sırasında kenarlığın hangi renkte olacağını ayarlayabilir.
   * Örnek Kullanım: **customBtn1.BorderCoIor = Color.Blue;**
4. **BackgroundColor (Arka Plan Rengi):**
   * Butonun arka plan rengini belirler.
   * Kullanıcı, tasarım sırasında butonun arka plan rengini istediği renge ayarlayabilir.
   * Örnek Kullanım: **customBtn1.BackgroundColor = Color.Yellow;**
5. **TextdColor (Metin Rengi):**
   * Buton metni rengini belirler.
   * Kullanıcı, tasarım sırasında buton metninin rengini istediği renge ayarlayabilir.
   * Örnek Kullanım: **customBtn1.TextdColor = Color.Black;**

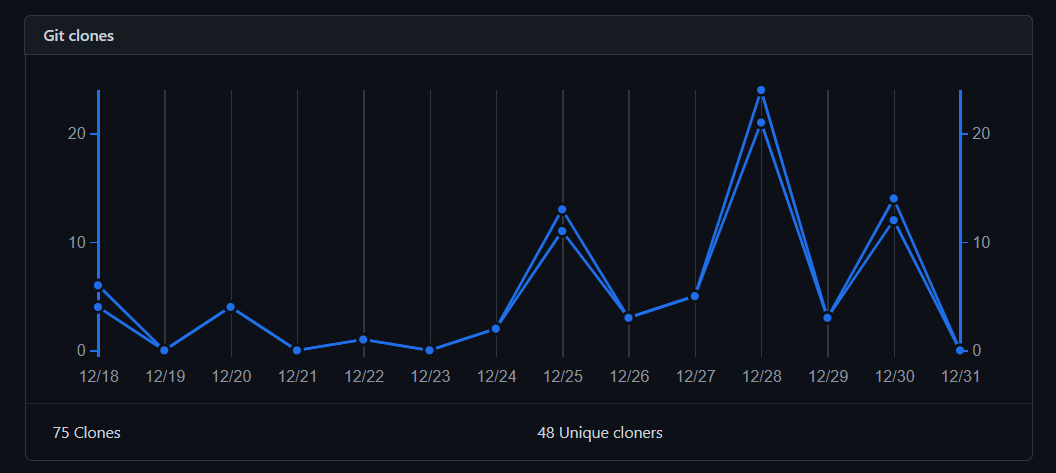
**CustomBtn Sınıfı Metotları**

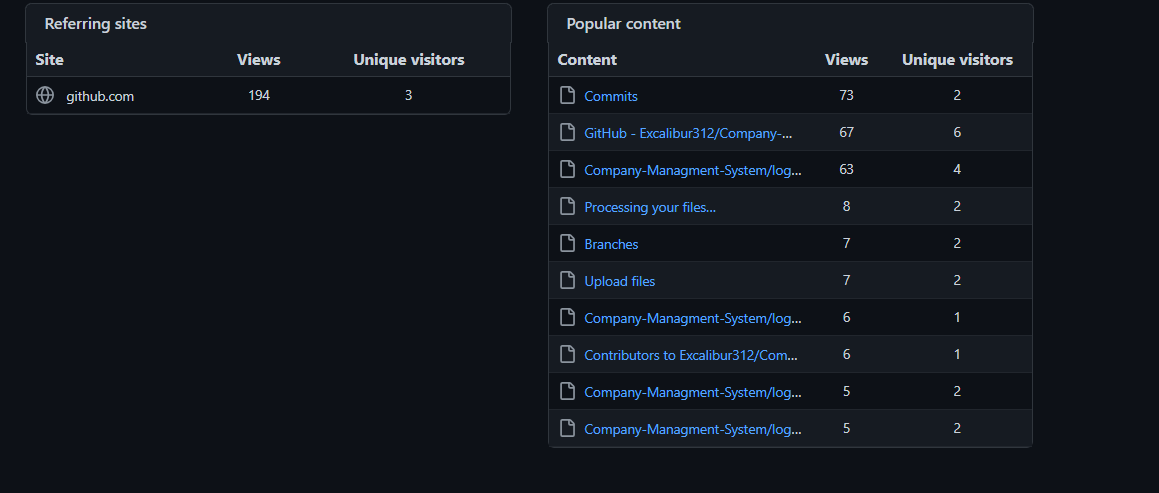
1. **GetFigurePath Metodu:**
   * Belirli bir yuvarlaklık derecesine göre yuvarlak kenarlı bir şeklin grafik yolunu döndürür.
2. **OnPaint Metodu:**
   * Butonun görünümünü özelleştirmek için kullanılır.
   * Kenar yarıçapına ve kenarlık kalınlığına göre düz veya yuvarlak bir buton oluşturur.
3. **OnHandleCreated Metodu:**
   * **Handle** özelliği oluşturulduğunda tetiklenir.
   * Ebeveynin arka plan rengi değiştiğinde butonu yeniden çizmek için bir olay ekler.
4. **Container\_BackColorChanged Metodu:**
   * Ebeveynin arka plan rengi değiştiğinde butonu yeniden çizen bir olay işleyicisidir.
5. **Button\_Resize Metodu:**
   * Buton boyutu değiştiğinde tetiklenir.
   * Kenar yarıçapı, buton yüksekliğinden büyükse düzeltilir.

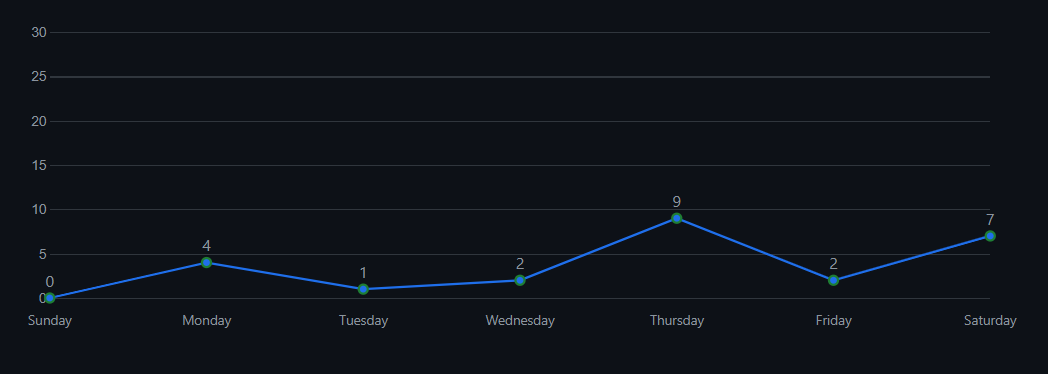
Bu özellikler ve metodlar sayesinde, her bir butonun tasarımını kolaylıkla özelleştirebilir ve kullanıcı dostu arayüzler oluşturabilirsiniz.

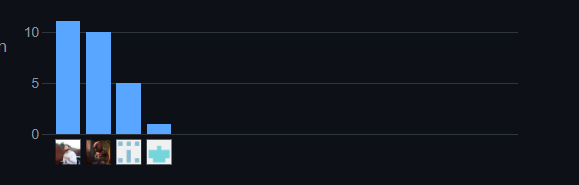
****

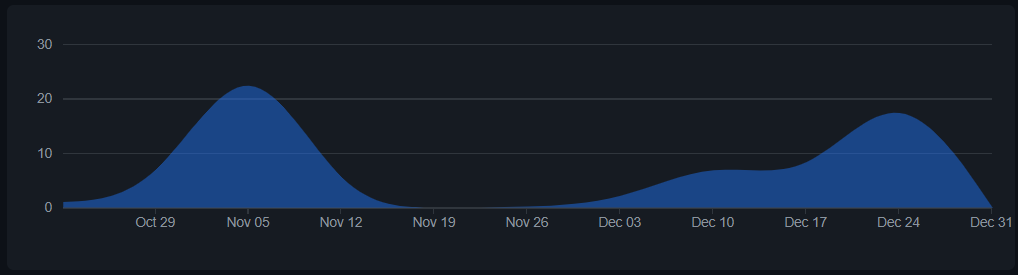
**Tablolar ve Grafikler**

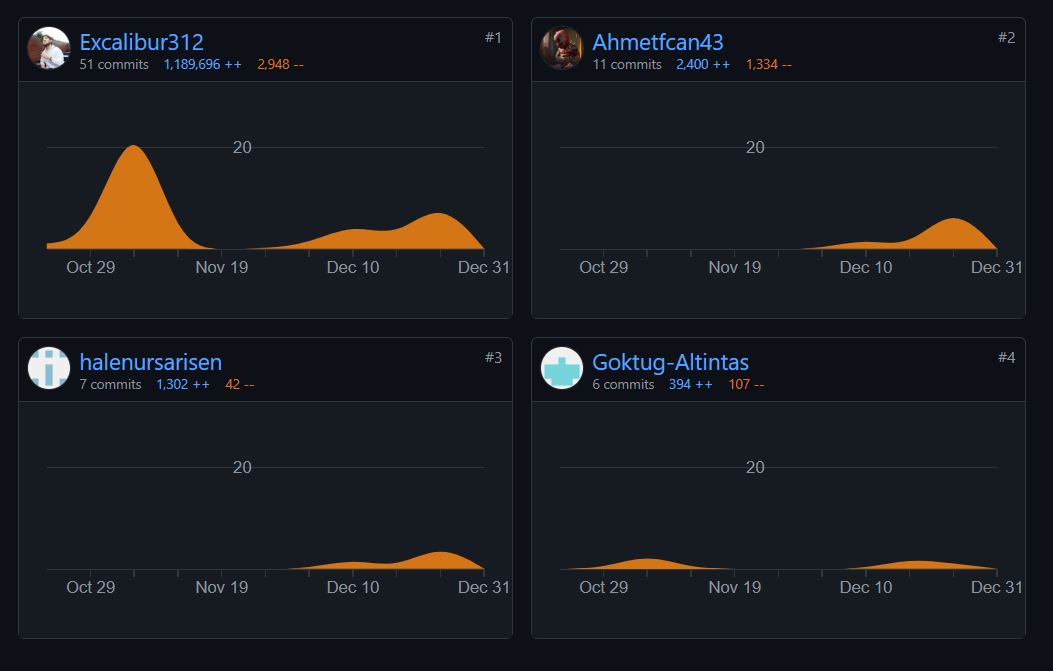
75 kere proje üzerinde çalışma yaptık****

75 kere commit ederek projede her işi yaparken detaylı ve titiz çalışarak projenin verimliliği artırdık. Her projenin her ufak detayını konuşarak beraber karar verdik ve muazzam projemizi oluşturduk. 

Her Perşembe son yaptıklarımızı commit ederek her Cuma ekip olarak ne yapacağımızı karar vermek için fiziksel toplantılar yaptık. 

Birleştirmeler hariç, 4 yazar ana 27 taahhüt ve tüm dallara 27 taahhüt gönderdi. Ana dosyada 386 dosya değişti ve 1.186.199 ekleme ve 499 silme işlemi gerçekleşti. 

Projemizi 2 ayda 8 haftada geliştirdik fakat vize haftalarımızdan kaynaklı projemizin 3 haftası çalışamadık ve 5 haftalık bir süreç gibi bir sürede geliştirdik. 



Commitlerimiz ve çalışma sıklığımız fakat şöyle bir şey var ilk haftalarda github’ı grupça çözerken baştaki commitler yönetici yapmıştır ama ilk aydaki commitleri 4’e bölündüğünü düşünersek 24 commit eder. Kısacası toplamda;  
Emre Kaya 26 commit   
Ahmet Fevzi Can 17 commit  
Hale Nur Sarışen 14 commit   
Göktuğ Altıntaş 12 commit edilmiştir