**Kütüphane Otomasyonu**

**Proje Raporu**

**Hazırlayan:**

**Necmeddin Yakup OKUMUŞ**

Projede Access Veri tabanı kullanıld. Veri tabanı dosyası KutuphaneOtomasyon0\KutuphaneOtomasyon0\bin\Debug içerisine eklendi.

Zedgraph için dll dosyası da aynı dizindedir.

**Otomasyonun İşleyişi**

**1-)Öğrenci İşlemleri**

**- (Öğrenci kayıt ekleme, güncelleme ve silme penceresi),** İlgili Formda 3 butın altında yapılan işlemler TC NO için ayrılan textBox kontrolününe göre yapılıyor yazılan TC NO veri tabanında varsa Öğrenci Ekleme Butonu Enabled özelliği false yapılıyor,TCNO karşılaştırma için ilgili textBoxa Text Changed Olayı yazılıyor. Veri tabanından sorguyla TC nolar çekiliyor, Girilen TCNO ile eşleşme varsa Güncelleme ve Silme İşlemleri butonu aktif edilip Ekleme pasifleşiyor.

**-Yapılan Ekleme, Silme ve Güncelleme** işlemlerinden sonra kullanılmak üzere textBox alanlarını temizleyen HepsiniSil() metodu yazılıyor kontollerin Text özelliklerine “” boş String atanıyor.

**-Öğrenci Ekleme**

TC NO alanı 11 haneliyse ve veri tabanındaki hiçbir kullanıcıyla eşleşmemişse ayrıca gerekli bütün bilgi alanları doldurulmuşsa işleme devam edilmesi if kontrolleriyle sağlanıyor.

İnsert sorgusu ile ilgili veriler veri tabanında okuyucu tablosuna kaydediliyor, daha sonra alanlar temizleniyor.( **HepsiniSil()** )

**-Öğrenci Güncelleme**

Eğer girilen **TCNO** veri tabanından herhangi bir değerle eşleşmişse buton aktif ediliyor ve Tüm gerekli alanların boş olmadığı kontrol ediliyor alanlarda kullanıcı güncelleyeceği bilgiyi değiştirip tuşlama yapınca **UPDATE** sorgusu ile ilgili veriler okuyucu tablosunda güncelleniyor.

**-Öğrenci Silme**

Girilen **TCNO** veri tabanında bulunmuşsa ilgili veriler otomatik alanları dolduruyor bundan sonra aktif olan güncelle ve silme butonlarından silme butonuna basılırsa **TCNO** baz alınarak veri tabanından “**delete from okuyucu where tcno** “sorgusu ile siliniyor ardından alanlar temizleniyor HepsiniSil().

**-Email Doğruluğu**

Girilen bilgilerden Email ile ilgili olan textBox için TxtEmailKontrol ismindeValidating olayı yazılıyor. Bu olayda girilen mailde **@** ve **nokta** işaretleri ile email formatına uygunluk kontrol ediliyor.

**2-)Kitap İşlemleri**

-**KitapId** yazılan textBox a **TextChanged** olayı yazılıyor. Veri tabanından ilgili İd için arama yapıyor. **SELECT** sorgusu ile veri tabanındaki bütün kayıtlı **Idlerler** girilen **ID** karşılaştırılıyor eşleşme varsa Text alanları veri tabanındaki verilerle doluyor yani kayıt ekrana getirilmiş oluyor. **Kitap Ekle** butonu pasif hale getiriliyor ardından. **Güncelle** ve **Silme** butonları aktif ediliyor.

**-Alanları temizlemek için HepsiniSİl()** isimli bir metot kullanılıyor, işlemlerden sonra ilgili alanların Text özelliklerine “” ataması yapılarak boşaltılmaları sağlanıyor.

**-Kitap Ekle**

ID uzunluğu doğru ise ve bütün alanlar dolu ise if kontrolüyle kontrol edilip **insert into (..........) values (........)** sorgusuyla veri tabanına ekleme işlemi yapılıyor. Daha sonra ilgili alanları temizleyen metot çalışıyor ( **HepsiniSil()** ) ardından işlem başında açılan veri tabanı bağlantısı kapatılıp işlem bitiriliyor.

**-Kitap Sil**

Girilen **KitapID** veri tabanında bulunduysa aktif hale gelen butonun altında yapılan işlemler;

Veri tabanı bağlantısı yapılıp **“delete from kitap where kitapid**” için sorgu yazılıyor, **DialogResult** ile kullanıcıdan onay alınırsa işlem yapılıp ilgili kitap veri tabanından siliniyor,ardından **HepsiniSil()** metodu dolu text alanlarını temizliyor ve veri tabanı bağlantısı kapatılıyor.

**-Kitap Güncelle**

**Kitap ID** veri tabanında bulununca ekrandaki **textBox** alanlarına veri tabanındaki bilgilerin çekilmesinin ardından **Güncelle** **butonu aktif** oluyor ve bu butonun altında yapılan işlemler;

Veri tabanı bağlantısı açılıyor **“update kitap set ...... where kitapid=”**

Sorgusu ile ilgili tüm değerler veri tabanında güncelleniyor, ardından veri tabanı bağlantısı kapatılıyor ve **HepsiniSil()** metodu ile ilgili alanların Text özellikleri temizleniyor.

**-Alanları Temizlemek** için ayrıca **buton** koyuluyor, bu buton altında tüm Text alnalrarını temizleyen **HepsiniSil()** metodu çalışıyor.

**3-) Kitap Al / Bırak**

**-İlgili formun Load** olayında **Öğrenci** ve **Kitapları** **listeleyen** veri tabanında veri çekme **sorgusu** çalışıyor ve ekrandaki tabloda listeleniyorlar,bu iki işlem sonunda veri tabanı bağlantıları kapatılıyor, Kullanıcı Öğrenci tablosunda seçmek için tıkladığında Tablonun altında bulunan **textbox** **alanları** tablodaki bilgilerden **doluyor**.Daha sonra Alınacak kitap kitapların olduğu **dataGridView** tablosundan seçilince kitap adi ve kitapıdsi ilgili **textBox** **kontrollerine** **yazılıyor**, daha sonra kitap kaydını kullanıcıya yaparken buradan veriler çekilecek.Ayrıca Kitaba tıklanınca Başlangıç tarihi yani sistemin o anki zamanı ve kitabın teslim tarihi için dateTimePicker kontrolleri görünür oluyor.

**-KitapKayit()** metodu çalışınca önce ilgili öğrencinin veri tabanındaki bilgiye ulaşarak elinde kaç kitap olduğunun kontrolü yapılıyor **en** **fazla** **5** kita alabilir her kullanıcı.**Bu kontrol if ile** yapılıp işlem devam ederse **insert into emanet** **SQL sorgusu** ile emanet tablosuna emanet için alınan bilgiler yazılıyor ardından **update kitap set kitapdurumu=** sorgusu ile **kitabın durumu 0** yapılıyor. Bu kitabın emanet verildiği elde bulunmadığı manasına geliyor. Ardından okuyucu tablosundan öğrencinin elinde bulunan kitap sayısı ve toplamda aldığı kitap sayısı select **select aldigikitapsayisi,emanetkitapsayisi from okuyucu where tcno like** sorgusu ile çekiliyor. Ve **değerler +1 işlemi yapılıp** veri tabanında **update** sorgusu ile güncelleniyor. İşlemlerin sonunda Ekrandaki Öğrencilerin listesini güncellemek için veri tabanından tüm veriler **SELECT** sorgusu ile çekilip ekrana listeleniyor.

**-Kitap Al Button:** bu buton veri tabanında ilgili kitabın id si ile **select** sorgusu yapılıyor eğer emanet tablosundan kitabın emanet alınıp alınmadığı kontrol ediliyor, kitabın emanet **kaydı yoksa KitapKayit()** metodu çağrılıyor ve işlem tamamlanıyor. Eğer kitabın emanet **kaydı varsa select kitapdurumu from kitap where kitapid like** sorgusu ile kitabın durumu kontrol ediliyor eğer **durum 0 ise** başkasına verilmiyor **eğer durum 1 ise** yani kitap kütüphanede ise **KitapKayit()** metodu çağrılıp işlem yapılıyor.

**-Öğrenci Ara:** İlgili **textBox** kontrolünün **TextChanged** olayına yazılan kodda veri tabanı bağlantısı açılıp girilen **TCNO** veri tabanında aranıyor uyuşan **TC’nin** bilgileri ekrana yazılıyor.

**-Kitap Ara:**İlgili **textBox** kontrolünün **TextChanged** olayına yazılan kodda veri tabanı bağlantısı açılıp **girilen KitapID** veri tabanında aranıyor **uyuşan KitapID’nin** bilgileri ekrana yazılıyor.

**-Kitap Teslim Et:** Butona tıklanınca bu işlem için yeni bir form açılıyor

**TCNO** **arama** kısmına ilgili tc ile ilgili birkaç rakam yazılıp kontrol butonu çalıştırılınca

**select E.adsoyad,E.tcno,E.kitapadi,E.durum,E.bitistarihi,E.kitapid,O.ceza from emanet E,okuyucu O where O.tcno=E.tcno and E.tcno like '" + Txt\_TCNO.Text + "%' and E.durum like '" + "Teslim Edilmedi" + "'** Sorgusu çalıştırılıyor.

Gelen değerler ekrandaki **dataGridView’e** yazılıyor.Veriler yazıldıktan sonra hemen kontrolden geçiyor, Teslim edilmesi planlanan tarih değeri alınıp Şuanki zamnala karşılaştırılıyor **DateTime tipinde** olan bu iki tarih verisinin **TimeSpan GunFarki = Suan.Subtract(Tarih);** ile arasındaki fark alınıp integera çevrildikten sonra **dataGridViewde** **Teslim tarihi geçtiyse Kırmızı** 2 gün veya **daha az kaldıysa Sarı** 2 günden **fazla süre varsa teslime Yeşil** yapılıyor arka planı.

**-Goster() metodu:**Yukarıdaki işlemle anı sorguyu kullanarak veri tabanındaki erileri çekip ekrana yazıyor ve ilgili renk değerlerinin atamasını **dataGridView** üzerinde gerçekliyor.Metodun amacı işlem yapıldıktan sonra çalıştırılıp Tablonun güncellenmesi.

-Ekrandaki tablodan kitap seçilince **kitabın ID** numarası Ekranda **visible** özelliği aktif edilen **textBox** kontrolüne yazılıyor ve **Onay** için bekleniyor.

**-KitapId** yazan textbox **boş değilse(if kontrolü)** veri tabanı bağlantısı açılıyor ve **update emanet set ...** sorgusu ile emanet tablosunda kitabın teslim tarihi güncelleniyor daha sonra **update kitap set** sorgusu ile kitabın durumu **güncelleniyor 1 yapılıyor**, kütüphanede bulunuyor olarak değişmiş oluyor durumu. Ardından **update emanet set durum=@durum where kitapid like** sorgusu ile kitabın durumu emanet tablosunda “Teslim Edildi” olarak güncelleniyor. Daha sonra kitabı alan kişinin elinde bulunan emanet sayısı **güncellenip 1 azaltılıyor update okuyucu set** **emanetkitapsayisi=@emanetsayisi Where TCNO Like** sorgusu ile. Ardından teslim edilen tarih ile teslim edilmesi gereken **tarih arasındaki fark alınıyor** **TimeSpan GunFarki = Suan.Subtract(Tarih);** bu farka göre veri tabanından okuyucu tablosundan ceza değeri çekilip güncelleniyor **update okuyucu set ceza=@ceza where tcno like**

**-Ardından Goster()** metodu çalışıyor ve veri tabanı bağlantısı kapatılıyor.

**4-) Emanet Kitaplar**

Bu sayfada öğrencilerin **şimdiye kadar almış oldukları** kitaplar ve **şu ana kadar alınan** **bütün kitaplar** listelenmektedir.

**Formun load** olayında **goster() metodu** çalışır ve ekrana öğrenciler aldıkları tüm kitaplarla birlikte listelenir.

**-goster()metodu** veri tabanından **Select** sorgusu ile emanet tablosunun verileri çekilerek ekrana **dataGridView** ile gösteriliyor.Ardından **dataGridView** kontrolünün bütün elemanları sayısı kadar for döngüsü içinde renlendiirme işlemi yapılıyor. **TimeSpan GunFarki = Suan.Subtract(Tarih);** koduyla **DateTime** tipinde olan Şuanki verimiz ayrıca **dataGridViewdeki** tarih bilgisi de **DateTİme** tipine **Convert** ediliyor ve **DateTime** tipinden değişkenlerde tutulan **bu iki değişkenin farkı alınıyor**,alınan fark **integera çevriliyor,**

Bu fark teslim tarihine kaç gün kaldığını yada geçip geçmediğini belirtiyor, **Teslim tarihine 2** **veya 0 arası** bir zaman kalmışsa **dataGridViewdeki** ilgili elemanın **arka planı Sarı** yapılıyor, eğer teslim tarihi aşıldıysa **arka plan Kırmızı** eğer 2 günden fazla varsa veya teslim edilmişse **arka plan rengi Yeşil atanıyor**.

**-TcNoileGoster() TcNo** ile aranan öğrencinin şimdiye kadar aldığı bütün kitaplar listeleniyor.

**Select adsoyad,tcno,kitapadi,bitistarihi,teslimtarihi,durum From emanet Where tcno LIKE** sorgusu ile emanet tablosundan tcno ile aranıyor. Gelen veriler **dataGridView** kontrolüne yazılarak kullanıcıya sunuluyor. Daha sonra **goster()** metodu için uygulanan renklendirme işlemi uygulanıyor. **TcNoileGoster()** metodu **Txt\_tcileSorgula** kontrolünün **TextChanged** olayına yazılıyor böylece her tuşlamada veri tabanından ilgili alana göre arama yapılıyor.

-**TeslimetmeynlerGoster()** emanet tablosundan durumu **“Teslim Edilmedi”** olan emanetlerin bilgileri sorguyla çekilip ekrana **dataGridView** ile yazılıyor. Ardından **renklendirme** işlemi **goster()** fonksiyonu ile aynı şekilde yapılıyor. Bu metot Teslimetmeyenler Göster Butonu altında çalışıyor.

**5-) Kitap Listele**

**KayitAl()** metodu **“Select kitapid,kitapadi,tur,yazar,yayinevi,basimyeri,basimtarihi,sayfasayisi from kitap “** sorgusu ile kitap tablosundan verileri çekiyor, Form ekranındaki **dataGridView** kontrolüne yazıyor.

Formun **load** olayında **KayitAl()** metodu çalıştırılıyor.

**Txt\_arama\_TextChanged** olayında ise **select \* from kitap where kitapid like ...** ile girilen kitapid ile veri tabanındaki ilgili veriler çekiliyor ve **dataGridViewle** ekrana yazılıyor.

**Btn\_teslimAlanlar\_Click** olayında ise emanet tablosundan yazılan kitapid için arama yapılıyor ve o kitabı teslim alan bütün okuyucular(**öğrenciler**) listeleniyor.

**6-)Öğrenci Listele**

-**kayital()** metodu okuyucu tablosundaki bütün verileri yani bütün öğrencileri **select \* from okuyucu** sorgusuyla alıp ekrana **dataGridView** kontrolüne yazıyor, işlem başında açılan veri tabanı bağlantısı kapatılıp işlem bitiriliyor.

-Formun **load** olayında **kayital()** metodu çağrılıyor ve ekran açıldığında veriler listeleniyor

**-textBox1\_TextChanged** olayı ise ekranda bulunan **textBox** kontrolünün arama işlemi yapması için yazılmıştır. Het tuşlamada veri tabanı bağlantısı açılır **select \* from okuyucu where tcno like...** sorgusu ile veri tabanından **tcno için arama** yapılır. Çekilen veriler **dataGridView** ile ekranda gösterilir.

**7-)Form1/Ana Ekran/ZedGrapg**

Ana ekran açıldığında **grafiğimizle** birlikte diğer işlemler için erişim sağlayan **butonlarla** birlikte gelir, o **butonların altında yazılan kod** ise şudur; her formun **bir nesnesi üretilir** ve üretilen nesne **.Show()** metodu ile ekranda gösterilir, böylece kullanıcı ilgili işlemin formuna ulaşmış olur, her form için yapılan kodlamalar kullanılan metotlar işlemlerin uygulanışı tek tek anlatılmıştır.

- **ZedGraph\_Olustur(ZedGraphControl zg)** Metodu veri tabanından toplam kitap sayısını **"Select count(\*) From kitap"** ve **durumu “Teslim Edildi”** olanların **sayısı Select count(\*) From emanet where durum Like** ve durumu **“Teslim Edilmedi”** olanların sayısı **Select count(\*) from emanet where durum like** sorgusu ile çekiliyor ve ilgili değişkenlerine atanıyorlar.

Daha sonra grafiğin **Ana başlığı x ekseni başlığı ve y ekseni** başlığı yazılıyor,ardından grafik için çekilen değerlerle **bar grafiğimiz oluşturuluyor** değişkenleri atanıyor.Barlar için özel **renkler atanıyor**.Grafik işlemleri tamamlanınca veri tabanı bağlantısı kapatılıp işlem bitiriliyor.Ve grafiğimiz açıldığında veri tabanında **tüm kitapların sayısını**, **teslim edilen** kitapların sayısını ve **teslim edilmeyenlerin** sayısını **gösterir** durumda oluyor.