

Okul İhtiyaç Karşılama Sistemi

Halil Dağhan Şengören

Bahadır Çolak

Mehmet Berk Yüksel

Alacaatlı Mahallesi Alacaatlı Cad.

Söğütözü Caddesi 22. Sokak no: 12

Söğütözü Caddesi 22. Sokak

4819/1 sok. Adapark Sitesi

TOBB ETU Öğrenci Konukevi

TOBB ETU Öğrenci Konukevi

Çayyolu/ANKARA

Çankaya/ANKARA

Çankaya/ANKARA

+905372209631

+905377843943

+905389221212

daghan.sengoren@gmail.com

bahdrcolak@gmail.com

mberk.yuksel@gmail.com

ÖZET

Bu raporda BİL 372 kodlu Veritabanı Sistemleri dersinde yapılmış proje anlatılacaktır.

1- Giriş

Projemiz bir okul yardım sistemidir. Bu uygulama, ihtiyaç sahibi okullar için kurulmuştur. Uygulamada okullar kendi ihtiyaçlarını bir sisteme tanımlayabilecekler böylece sisteme kayıtlı olan yardımsever insanlar ihtiyaç sahibi bu okulların ihtiyaçlarını hızlı ve kesin olarak görebilecekler. Dolayısıyla okulların ihtiyaç fazlası ya da gereksiz ürünlerin alması engellenecektir. Sisteme kayıtlı olan her bir kullanıcının (adminler hariç) bakiyeleri bulunacaktır. Bu bakiyeler sayesinde yardımseverler okulların yayınladıkları ihtiyaç listelerini tedarikçilerin belirlediği fiyatlar doğrultusunda direk olarak karşılayabileceklerdir. Yardımseverler aynı zamanda para bağışında da bulunabilecekler böylelikle, bağışlanan para okulun bakiyesine gidecek ve okullar kendi ihtiyaçlarını kendileri karşılayabilecekler. Sistemde az bağış almış okullar yardımseverlere üst sıralarda görünecek fakat yardımseverler okulları aratarak da bulabileceklerdir. Veritabanımız Microsoft Azure üzerinde, Microsoft'un Kuzey Avrupa serverlarında POSTGRESQL desteğiyle maksimum güvenlik ve verimle çalışmaktadır. Uygulamamız C# Windows Form Application kullanarak Visual Studio üzerinde bir masaüstü uygulaması olarak hazırlanmıştır.

2- GENEL BAKIŞ

Bu bölümde C# Windows Form Uygulamasıyla yaptığımız uygulamanın sınıfları, alt sınıfları ve fonksiyonları ile ilgili genel bir bilgi verilecektir.

1. Sınıflar:

- 1.1. Login.cs
- 1.2. Register.cs
- 1.3. Yardımsever
 - 1.3.1. ihtiyac_karsila.cs
 - 1.3.2. OkulIhtiyac.cs
 - 1.3.3. para_bagisla.cs
 - 1.3.4. para_yukle.cs
 - 1.3.5. yardımseverpages.cs
 - 1.3.6. yardımsever2.cs

1.4. Tedarikci

- 1.4.1. guncelle.cs
- 1.4.2. tedarikci_pages.cs
- 1.4.3. urun_cikar.cs
- 1.4.4. urun_ekle.cs
- 1.4.5. urun_guncelle.cs

1.5. Okul

- 1.5.1. İhtiyaçlar
 - 1.5.1.1. Giysi.cs
 - 1.5.1.2. Kırtasiye.cs
 - 1.5.1.3. Kitap.cs
 - 1.5.1.4. Mobilya.cs
 - 1.5.1.5. Teknoloji.cs

- 1.5.2. Geçmiş.cs
- 1.5.3. guncelle.cs
- 1.5.4. ihtiyac_ekle.cs
- 1.5.5. ihtiyac_guncelle.cs
- 1.5.6. ihtiyac_karsila.cs
- 1.5.7. okul_pages.cs

1.6. admin

- 1.6.1. istek_sayfaları
 - 1.6.1.1. okul_istek.cs
 - 1.6.1.2. tedarikci_istek.cs
- 1.6.2. silme_sayfaları
 - 1.6.2.1. okul_sil.cs
 - 1.6.2.2. tedarikci_sil.cs
 - 1.6.2.3. yardımsever_sil.cs
- 1.6.3. admin_pages.cs

2. Functions

2.1. Login()- construction

- 2.1.1. button1_Click
- 2.1.2. button2_Click

2.2. Register()-construction

- 2.2.1. button2_Click
- 2.2.2. yardımsever_kayit_Click
- 2.2.3. IsValidEmail
- 2.2.4. Checkbox
- 2.2.5. checkBox1_CheckedChanged
- 2.2.6. checkBox2_CheckedChanged
- 2.2.7. button4_Click
- 2.2.8. okul_kayit
- 2.2.9. tedarikci_kayit_Click

2.3. ihtiyac_karsila() – construction

- 2.3.1. ihtiyac_karsila_Load
- 2.3.2. kayitGetir
- 2.3.3. listeye_ekle_Click

- 2.3.4. button1_Click
- 2.3.5. karsila_Click
- 2.3.6. deneme
- 2.4. Okullhtiyac() – construction
 - 2.4.1. dataGridView1_CellContentClick
 - 2.4.2. kayitGetir
 - 2.4.3. Okul_Load
 - 2.4.4. Geri_Click
 - 2.4.5. para_bagisla_Click
 - 2.4.6. button1_Click
- 2.5. para_bagisla() – construction
 - 2.5.1. Geri_Click
 - 2.5.2. kayitGuncelle
 - 2.5.3. bagisla_Click
 - 2.5.4. bakiyeGetir
 - 2.5.5. para_bagisla_Load
- 2.6. para_yukle() – construction
 - 2.6.1. geri_Click
 - 2.6.2. bakiye_yukle_Click
 - 2.6.3. kayitGuncelle
- 2.7. yardımsever_pages() – construction
 - 2.7.1. button1_Click
 - 2.7.2. yardımsever1_Load
 - 2.7.3. kayitGetir
 - 2.7.4. button3_Click
 - 2.7.5. bakiye_btn_Click
- 2.8. yardımsever2 – construction
 - 2.8.1. Form1_Load
 - 2.8.2. kayitGetir
 - 2.8.3. goto_okulpage
 - 2.8.4. button3_Click
 - 2.8.5. okul_ara_Click
- 2.9. guncelle() – construction
 - 2.9.1. button1_Click
 - 2.9.2. button2_Click
- 2.10. tedarikci_pages – construction
 - 2.10.1. ekle_btn_Click
 - 2.10.2. cikar_btn_Click
 - 2.10.3. update_btn_Click
- 2.11. urun_cikar – construction
 - 2.11.1. geri_btn_Click
 - 2.11.2. sil_btn_Click
 - 2.11.3. kayitGetir
 - 2.11.4. urun_cikar_Load
- 2.12. urun_ekle
 - 2.12.1. geri_btn_Click
 - 2.12.2. ekle_btn_Click
 - 2.12.3. clear
 - 2.12.4. urun_ekle_Load
 - 2.12.5. kayitGetir
- 2.13. urun_guncelle – construction
 - 2.13.1. kayitGetir
 - 2.13.2. urun_guncelle_Load
 - 2.13.3. button2_Click
 - 2.13.4. Guncelle_Click
 - 2.13.5. button1_Click
- 2.14. giysi – construction
 - 2.14.1. fiyatHesapla
 - 2.14.2. giysi_ekle_Click

- 2.14.3. giysi_geri_Click
- 2.14.4. kayitGetir
- 2.14.5. giysi_Load
- 2.14.6. comboboxGetir
- 2.14.7. totalfiyatgetir
- 2.15. kirtasiye – construction
 - 2.15.1. fiyatHesapla
 - 2.15.2. button2_Click
 - 2.15.3. button1_Click
 - 2.15.4. clear
 - 2.15.5. kayitGetir
 - 2.15.6. kirtasiye_Load
 - 2.15.7. comboboxGetir
 - 2.15.8. totalfiyatgetir
- 2.16. kitap – construction
 - 2.16.1. fiyatHesapla
 - 2.16.2. button2_Click
 - 2.16.3. button1_Click
 - 2.16.4. clear
 - 2.16.5. kayitGetir
 - 2.16.6. totalfiyatgetir
 - 2.16.7. kitap_Load
 - 2.16.8. comboboxGetir
- 2.17. mobilya – construction
 - 2.17.1. mobilya_geri_Click
 - 2.17.2. fiyatHesapla
 - 2.17.3. mobilya_ekle_Click
 - 2.17.4. clear
 - 2.17.5. kayitGetir
 - 2.17.6. totalfiyatgetir
 - 2.17.7. mobilya_Load
 - 2.17.8. comboBoxGetir
- 2.18. teknoloji – construction
 - 2.18.1. teknoloji_Load
 - 2.18.2. fiyatHesapla
 - 2.18.3. comboboxGetir
 - 2.18.4. teknoloji_geri_Click
 - 2.18.5. kayitGetir
 - 2.18.6. totalfiyatgetir
 - 2.18.7. teknoloji_ekle
 - 2.18.8. clear
- 2.19. geçmiş – construction
 - 2.19.1. gecmis_Load
 - 2.19.2. button1_Click
 - 2.19.3. kayitGetir
- 2.20. güncelle – construction
 - 2.20.1. button1_Click
 - 2.20.2. button2_Click
- 2.21. ihtiyac_ekle – construction
 - 2.21.1. button1_Click
 - 2.21.2. button3_Click
 - 2.21.3. button2_Click
 - 2.21.4. button4_Click
 - 2.21.5. button5_Click
 - 2.21.6. button6_Click
- 2.22. ihtiyac_guncelle – construction
 - 2.22.1. ihtiyac_guncelle_load
 - 2.22.2. kayitGetir
 - 2.22.3. button1_Click

- 2.22.4.button2_Click
- 2.22.5.Guncelle_Click
- 2.23. ihtiyac_karsila – construction
 - 2.23.1.ihtiyac_karsila_Load
 - 2.23.2.listeye_ekle_Click
 - 2.23.3.button1_Click
 - 2.23.4.kayitGetir
 - 2.23.5.karsila_Click
 - 2.23.6.deneme
- 2.24. okul_pages– construction
 - 2.24.1.button2_Click
 - 2.24.2.okul_pages_Load
 - 2.24.3.kayitGetir
 - 2.24.4.button1_Click
 - 2.24.5.button3_Click
 - 2.24.6.button4_Click
- 2.25. okul_istek– construction
 - 2.25.1.okul_istek
 - 2.25.2.okul_istek_Load
 - 2.25.3.kayitGetir
 - 2.25.4.onay_click_Click
 - 2.25.5.geri_btn_Click
 - 2.25.6.sil_btn_Click
- 2.26. tedarikci_istek– construction
 - 2.26.1.onayla_btn_Click
 - 2.26.2.geri_btn_Click
 - 2.26.3.tedarikci_istek_Load
 - 2.26.4.kayitGetir
 - 2.26.5.sil_btn_Click
- 2.27. okul_sil– construction
 - 2.27.1.okul_sil_Load
 - 2.27.2.kayitGetir
 - 2.27.3.button1_Click
 - 2.27.4.button2_Click
- 2.28. tedarikci_sil– construction
 - 2.28.1.button2_Click
 - 2.28.2.button1_Click
 - 2.28.3.tedarikci_sil_Load
 - 2.28.4.kayitGetir
- 2.29. yardımsever_sil– construction
 - 2.29.1.yardımsever_sil_Load
 - 2.29.2.kayitGetir
 - 2.29.3.button2_Click
 - 2.29.4.button1_Click
- 2.30. admin_pages– construction
 - 2.30.1.admin_pages
 - 2.30.2.button2_Click
 - 2.30.3.button3_Click
 - 2.30.4.button5_Click
 - 2.30.5.button4_Click

3- İşleyiş

Programımızı ilk çalıştırdığımızda karşımıza çıkan ilk ekran Login sayfasıdır. Her bir kullanıcı bu sayfadan giriş yapmak zorundadır.

Kullanıcılar giriş yaptıklarında veri tabanında tutulan int türündeki yetki, kullanıcıların hangi sayfaya yönlendirileceğine karar vermektedir.

Yardımcı -1	Okul -2
Tedarikçi -3	Admin -4

Girişi sağlayan SQL sorgusu:

```
NpgsqlCommand cmd = new
NpgsqlCommand("Select yetki from kullanıcı
where kullanıcı_adi=@kullanıcı_adi and
sifre=@sifre", conn);
```

Burda @ injection'ı engellemek amaçlıdır ve c# da ki kullanımı:
cmd.Parameters.AddWithValue("@kullanıcı_adi", kullanıcı_adi.Text);

Eğer Register butonuna tıklanırsa aşağıdaki gibi bir ekran açılacaktır:

Burda Yardım sever kayıt ekranı vardır. Kayıt etmeden önce valid mail kontrolü yapmaktadır ve tüm alanların doldurulmasını istemektedir.
Kayıt sorgusu:

```
NpgsqlCommand cmd = new NpgsqlCommand("insert
into
kullanıcı(kullanıcı_adi,sifre,email,il,ilce,mahalle,adres,p
osta_kodu,telefon,yetki)" +
" values(@kullanıcı_adi, @sifre,
@email, @il, @ilce, @mahalle, @adres, @posta,
@telefon ,1)", conn);
```

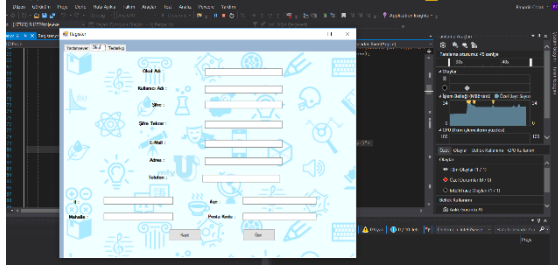
Ve

```
NpgsqlCommand cmd2 = new NpgsqlCommand("insert
into yardımsever(kAdi,ad,soyad,cinsiyet,bakiye)" +
```

```
" values(@kAdi, @ad, @soyAd, @cinsiyet, 0)", conn );
```

Şeklinde önce kullanıcı tablosuna ardından yardımsever tablosuna ekleme yapmaktadır.

Okul Register Sayfası:



Kayıt etmeden önce valid kontrolünü aşağıdaki SQL sorgusu yapmaktadır.

```
NpgsqlCommand cmd2 = new NpgsqlCommand("Select  
valid from okul where kullanıcı_adi=@kullanıcı_adi",  
conn);
```

Tüm alanların doldurulmasını istemektedir. Kayıt sorgusu:

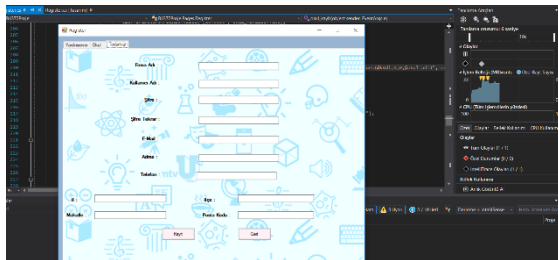
```
NpgsqlCommand cmd = new NpgsqlCommand("insert  
into  
kullanici(kullanici_adi,sifre,email,il,ilce,mahalle,adres,p  
osta_kodu,telefon,yetki)" + " values(@kullanici_adi,  
@sifre, @email, @il, @ilce, @mahalle, @adres,  
@posta, @telefon ,2)", conn);
```

ve

```
NpgsqlCommand cmd2 = new NpgsqlCommand("insert  
into okul(kAdi,bakiye,valid,okul_adi)  
values(@kAdi,0,0,@okul_adi)", conn);
```

Okul kaydı yapılırken, en son iki alan 0,0 olarak görünür bunlar biri bakiye diğeri valid. Burada okulun geçerliliği adminler tarafından kontrol edilinceye kadar 0 olarak duracaktır.

Son kaydımız, Tedarikçiye



Kayıt etmeden önce valid mail kontrolü yapmaktadır ve tüm alanların doldurulmasını istemektedir. Kayıt sorgusu:

```
NpgsqlCommand cmd = new NpgsqlCommand("insert  
into  
kullanici(kullanici_adi,sifre,email,il,ilce,mahalle,adres,p  
osta_kodu,telefon,yetki)" + " values(@kullanici_adi,  
@sifre, @email, @il, @ilce, @mahalle, @adres,  
@posta, @telefon ,3)", conn);
```

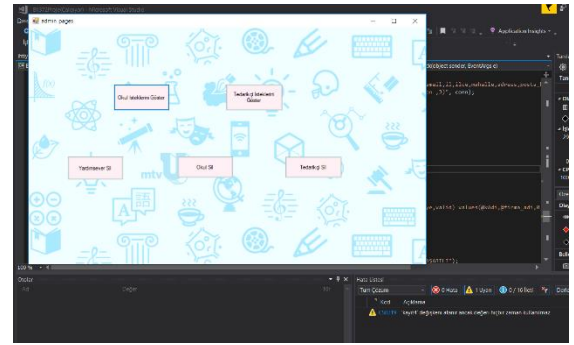
ve

```
NpgsqlCommand cmd2 = new NpgsqlCommand("insert  
into tedarikci(kAdi,firma_adi,bakiye,valid)  
values(@kAdi,@firma_adi,0,0)", conn);
```

Giriş yapıldıktan sonra;

Admin sayfaları:

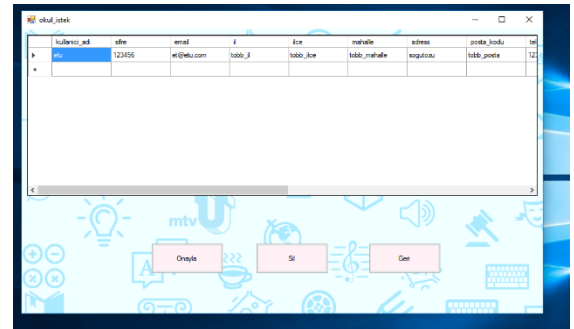
Sisteme giriş yaptığınızda eğer yetki türünüz 4 ise admin sayfasına yönlendiriliyorsunuz. İlk Sayfa:



Burada okul isteklerini göre tıkladığımızda aşağıdaki sayfa karşımıza gelecektir. Bu sayfaya yönlendirme kodu aşağıdaki gibidir:

```
okul_istek okul = new okul_istek();  
this.Hide();  
okul.Show();
```

Sayfamız açıldığında karşımıza gelen sayfa:



Burada okul tablosunda valid değeri 0 olanlar listelenmektedir. Listeleme SQL kodu:

```
"SELECT * " + "from kullanıcı k" + "inner join okul o on  
o.valid = 0 and kullanıcı_adi=o.kAdi";
```

Eğer admin isteği onaylayacaksa Onayla butonuna basmalıdır. Burdan sonra çalışacak SQL kodu:

```
NpgsqlCommand cmd = new NpgsqlCommand("update okul set valid = 1 where kAdi=@kullanici_adi", conn);
```

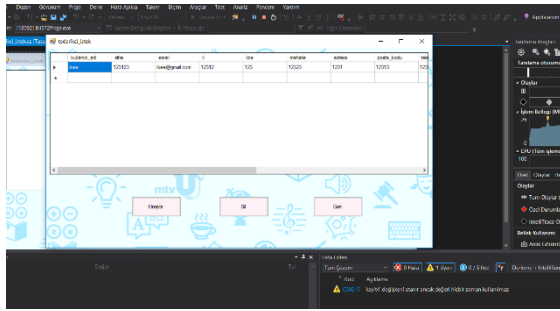
Eğer admin onaylamayacaksa sil butonuna basması gerekmektedir. Bu butona basıldığında yapılacak SQL sorgusu:

```
NpgsqlCommand cmd = new NpgsqlCommand("delete from okul where kAdi=@kullanici_adi", conn);
```

ve

```
NpgsqlCommand cmd2 = new NpgsqlCommand("delete from kullanıcı where kullanıcı_adi=@kullanici_adi", conn);
```

Tedarikçi istek Sayfası:



Eğer admin isteği onaylayacaksa Onayla butonuna basmalıdır. Burdan sonra çalışacak SQL kodu:

```
NpgsqlCommand cmd = new NpgsqlCommand("update tedarikci set valid = 1 where kAdi=@kullanici_adi", conn);
```

Eğer admin onaylamayacaksa sil butonuna basması gerekmektedir. Bu butona basıldığında yapılacak SQL sorgusu:

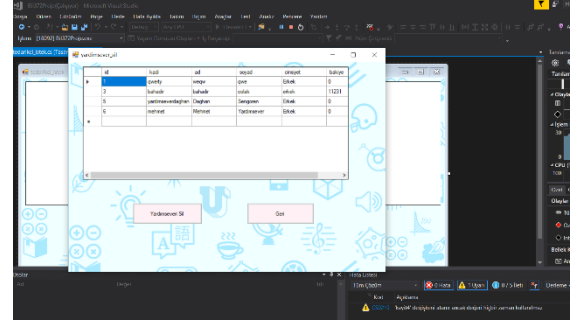
```
NpgsqlCommand cmd = new NpgsqlCommand("delete from tedarikci where kAdi=@kullanici_adi", conn);
```

ve

```
NpgsqlCommand cmd2 = new NpgsqlCommand("delete from kullanıcı where kullanıcı_adi=@kullanici_adi", conn);
```

Yardımsever sil sayfası:

Admin bu butona bastığında karşısına resimdeki gibi bir ekran gelir ve var olan kullanıcılar listelenir:



Listeleme SQL sorgusu:

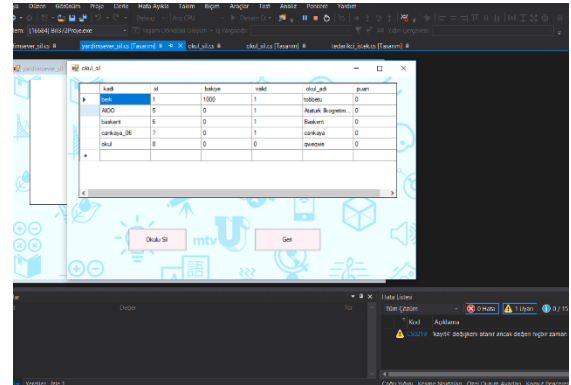
```
String yardımsever = "select * from yardımsever";  
NpgsqlCommand komut = new  
NpgsqlCommand(yardımsever, conn);
```

Kullanıcı sil butonuna bastığında çalışan sorgu:

```
NpgsqlCommand cmd = new NpgsqlCommand("delete from yardımsever where kAdi=@kullanici_adi", conn);  
NpgsqlCommand cmd2 = new NpgsqlCommand("delete from kullanıcı where kullanıcı_adi=@kullanici_adi", conn);
```

Okul sil sayfası:

Admin bu butona bastığında karşısına resimdeki gibi bir ekran gelir ve var olan okullar listelenir:



Listeleme SQL sorgusu:

```
String yardımsever = "select * from okul";  
NpgsqlCommand komut = new  
NpgsqlCommand(yardımsever, conn);
```

Okul sil butonuna bastığında çalışan sorgu:

```
NpgsqlCommand cmd = new NpgsqlCommand("delete from okul where kAdi=@kullanici_adi", conn);
```

ve

The screenshot shows a Windows desktop with a web browser window open. The browser displays a table with the following data:

id	isim	sinir_adi	bulunise	valdi
1	apin	apin	12312	1
2	apin_25	Apin	0	1
4	inaz,jarick	NEA	0	1
5	inaz	inaz	0	0

Below the table, there are two buttons: "Tutukla gür 50" and "Geri". The taskbar at the bottom shows the Start button, a search bar, and several open applications including a file explorer, a web browser, and a terminal window.

The screenshot shows a Visual Studio Code editor window with a C# project. The 'Program.cs' file is open, displaying the following code:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace CSVReader
{
    class Program
    {
        static void Main(string[] args)
        {
            string csvFile = "data.csv";
            Console.WriteLine($"Reading CSV file: {csvFile}");
            using (var reader = new StreamReader(csvFile))
            {
                var lines = reader.ReadToEnd().Split("\n");
                foreach (var line in lines)
                {
                    Console.WriteLine(line);
                }
            }
        }
    }
}
```

The output window shows the following output:

```
Reading CSV file: data.csv
1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,242,243,244,245,246,247,248,249,250,251,252,253,254,255,256,257,258,259,260,261,262,263,264,265,266,267,268,269,270,271,272,273,274,275,276,277,278,279,280,281,282,283,284,285,286,287,288,289,290,291,292,293,294,295,296,297,298,299,300,301,302,303,304,305,306,307,308,309,310,311,312,313,314,315,316,317,318,319,320,321,322,323,324,325,326,327,328,329,330,331,332,333,334,335,336,337,338,339,340,341,342,343,344,345,346,347,348,349,350,351,352,353,354,355,356,357,358,359,360,361,362,363,364,365,366,367,368,369,370,371,372,373,374,375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,398,399,400,401,402,403,404,405,406,407,408,409,410,411,412,413,414,415,416,417,418,419,420,421,422,423,424,425,426,427,428,429,430,431,432,433,434,435,436,437,438,439,440,441,442,443,444,445,446,447,448,449,450,451,452,453,454,455,456,457,458,459,460,461,462,463,464,465,466,467,468,469,470,471,472,473,474,475,476,477,478,479,480,481,482,483,484,485,486,487,488,489,490,491,492,493,494,495,496,497,498,499,500,501,502,503,504,505,506,507,508,509,510,511,512,513,514,515,516,517,518,519,520,521,522,523,524,525,526,527,528,529,530,531,532,533,534,535,536,537,538,539,540,541,542,543,544,545,546,547,548,549,550,551,552,553,554,555,556,557,558,559,560,561,562,563,564,565,566,567,568,569,570,571,572,573,574,575,576,577,578,579,580,581,582,583,584,585,586,587,588,589,590,591,592,593,594,595,596,597,598,599,600,601,602,603,604,605,606,607,608,609,610,611,612,613,614,615,616,617,618,619,620,621,622,623,624,625,626,627,628,629,630,631,632,633,634,635,636,637,638,639,640,641,642,643,644,645,646,647,648,649,650,651,652,653,654,655,656,657,658,659,660,661,662,663,664,665,666,667,668,669,670,671,672,673,674,675,676,677,678,679,680,681,682,683,684,685,686,687,688,689,690,691,692,693,694,695,696,697,698,699,700,701,702,703,704,705,706,707,708,709,710,711,712,713,714,715,716,717,718,719,720,721,722,723,724,725,726,727,728,729,730,731,732,733,734,735,736,737,738,739,740,741,742,743,744,745,746,747,748,749,750,751,752,753,754,755,756,757,758,759,760,761,762,763,764,765,766,767,768,769,770,771,772,773,774,775,776,777,778,779,780,781,782,783,784,785,786,787,788,789,790,791,792,793,794,795,796,797,798,799,800,801,802,803,804,805,806,807,808,809,810,811,812,813,814,815,816,817,818,819,820,821,822,823,824,825,826,827,828,829,830,831,832,833,834,835,836,837,838,839,840,841,842,843,844,845,846,847,848,849,850,851,852,853,854,855,856,857,858,859,860,861,862,863,864,865,866,867,868,869,870,871,872,873,874,875,876,877,878,879,880,881,882,883,884,885,886,887,888,889,890,891,892,893,894,895,896,897,898,899,900,901,902,903,904,905,906,907,908,909,910,911,912,913,914,915,916,917,918,919,920,921,922,923,924,925,926,927,928,929,930,931,932,933,934,935,936,937,938,939,940,941,942,943,944,945,946,947,948,949,950,951,952,953,954,955,956,957,958,959,960,961,962,963,964,965,966,967,968,969,970,971,972,973,974,975,976,977,978
```

Bu işlemleri yapan SQL sorguları:


```
string kayit = "insert into
gecmis(okul_id,yardimsever_id,ad,marka,adet,fiyat,tur)
values(@okul_id,1,@ad,@marka,@adet,@fiyat,@tur)";
```

ve

```
string kayit2 = "delete from " +
listView1.Items[i].SubItems[5].Text + " where ihtiyac_id
= @ihtiyac_id";
```

```
kayit3 = "delete from ihtiyac where id = @ihtiyac_id";
```

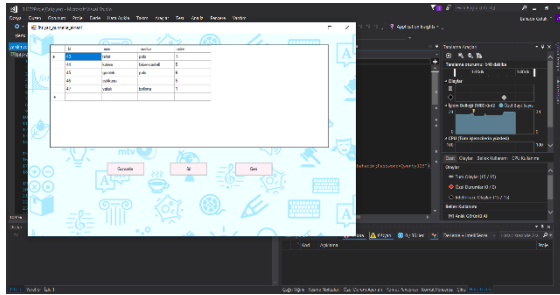
veya

```
kayit3 = "update ihtiyac set adet=" + kalanAdet + "
where id = @ihtiyac_id";
```

ve

```
kayit4 = "update yardımsever set bakiye=bakiye-@price
where kullanıcıAdi=@kullanıcıAdi";
```

Diğer bir sayfamız ihtiyaç güncelle sayfasıdır.

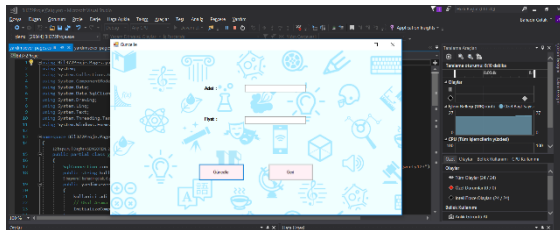


Görüldüğü üzere tablodan seçilen bir satır üzerinde iki farklı işlem yapılır. Bunlar güncelle ve sil işlemidir.

Sil işlemi yapıldığı zaman çalışan SQL sorgusu:

```
NpgsqlCommand cmd = new NpgsqlCommand("delete
from ihtiyac where Id=@Id", conn);
```

Diğer bir işlem olan güncelle butonuna tıklandığında ise aşağıdaki güncelle sayfasına gidilir.



Burada güncellenmek istenen adet ve fiyat alanları değiştirilir. Güncelle butonuna tıklandığı zaman öncelikle

kutuların boş olup olmadığı kontrol edilir. Boş ise mesaj kutusu ile kullanıcı uyarılır eğer değilse şu SQL sorgusu çalışır:

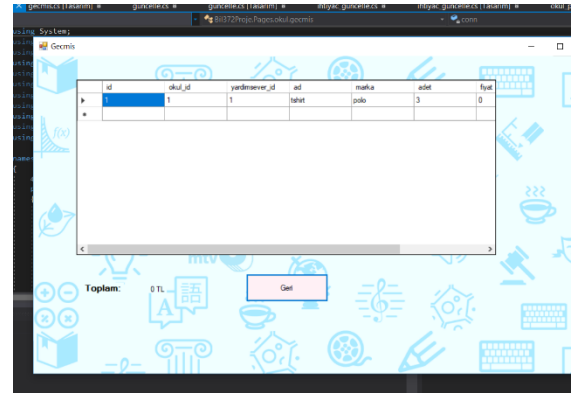
```
string kayit = "update ihtiyac set adet=@adet where
Id=@Id";
```

```
NpgsqlCommand komut = new
NpgsqlCommand(kayit, conn);
```

```
string kayit2 = "update ihtiyac set fiyat=@fiyat where
Id=@Id";
```

```
NpgsqlCommand komut2 = new
NpgsqlCommand(kayit2, conn);
```

Bir diğer sayfa ise geçmiş sayfasıdır.



Okul ana sayfasındayken geçmiş butonuna tıkladığımızda bu sayfa açılır. Sayfada geçmiş yardımlar ve bunların toplam değeri gözüktür. Bu butona tıklandığı zaman arka tarafta çalışan SQL sorgusu şu şekildedir:

```
string kayit = "SELECT SUM(fiyat) " +
"from gecmis";
```

```
string kayit2 = "SELECT * " +
"from gecmis";
```

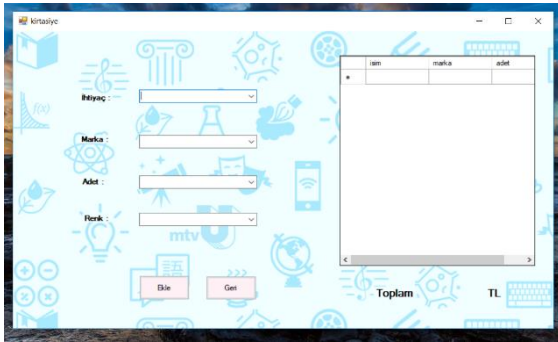
```
NpgsqlCommand komut2 = new
NpgsqlCommand(kayit, conn);
NpgsqlCommand komut = new
NpgsqlCommand(kayit2, conn);
```

Okul ana sayfasındayken ihtiyaç ekle butonuna tıklandığında ihtiyaç sayfasına gidilir. Burada gözüktüğü gibi beş farklı kategori bulunmaktadır. Bunlar:

Giysi ihtiyacı ekle, Kitap ihtiyaçları ekle, Teknoloji ihtiyaçları ekle, Kırtasiye ihtiyaçları ekle, Mobilya ihtiyaçları ekle butonlarıdır. Aşağıdaki sayfada görülmektedir.



İlk olarak Kırtasiye ihtiyaçları ekle butonuna tıkladığımızda aşağıdaki Kırtasiye sayfası açılır.



Bu sayfada ekle butonuna basıldığında ilk olarak toplam fiyatın hesaplandığı bir SQL sorgusu daha sonra ise doldurulması gereken alanlar doldurulduğunda çalışan bir SQL sorgusu vardır. Bunlar:

```
NpgsqlCommand kontrol = new NpgsqlCommand("select
adet from ihtiyac where isim=@isim and
marka=@marka", conn);
```

Eğer daha önceden kayıtlı ise aşağıdaki güncelleme sorgusu çalışır.

```
NpgsqlCommand cmd = new NpgsqlCommand("Update
ihtiyac set adet=@Adet Where isim = @isim", conn);
NpgsqlCommand cmd3 = new
NpgsqlCommand("Update ihtiyac set fiyat=@fiyat
Where isim = @isim", conn);
```

Değilse yeni bir ekleme yapılır.

```
NpgsqlCommand cmd = new NpgsqlCommand("insert
into ihtiyac(okul_id,isim,adet,marka,fiyat,tur)" +
"
values(@@okul_id,@isim,@adet,@marka,@fiyat,'kirtasiye
')", conn);
```

Daha sonra kırtasiye tablosuna eklenir.

```
NpgsqlCommand komut = new NpgsqlCommand("select
max(id) from ihtiyac", conn);
```

```
NpgsqlCommand cmd2 = new NpgsqlCommand("insert
into kirtasiye(ihtiyac_id)" +
" values(@ihtiyac_id)", conn);
```

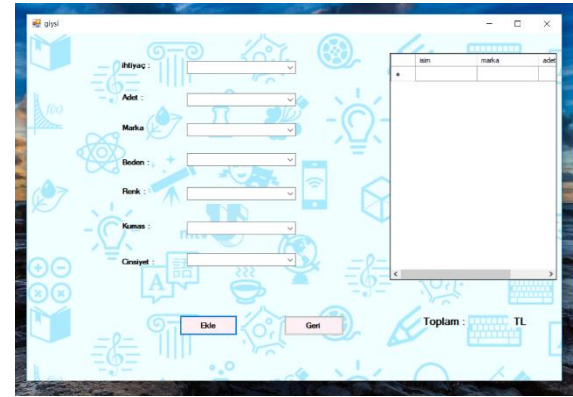
Buradaki sorguda ise ekrana okula ait olan ihtiyaçlar getirilir.

```
string kayit = "SELECT isim,marka,adet " +
" from ihtiyac where ihtiyac.okul_id
=@okul_id";
```

Buradaki sorguda ise combobox'lara kategoriye ait olan ürünler getirilir.

```
komut.CommandText = "SELECT * FROM urun where
kategori = @kategori";
komut.Parameters.AddWithValue("@kategori",
"kirtasiye");
```

Bir önceki sayfaya gelip giysi ihtiyaçları ekle butonuna bastığımız zaman ise aşağıdaki giysi sayfası açılır.



Bu sayfada çalışan SQL sorguları şu şekildedir:

```
NpgsqlCommand kontrol = new NpgsqlCommand("select
adet from ihtiyac where isim=@isim and
marka=@marka", conn);
Bu sorgu ile fiyat hesaplanıyor.
```

```
NpgsqlCommand kontrol2 = new
NpgsqlCommand("select ihtiyac_id from giysi where
beden=@beden and renk=@renk and kumas=@kumas
and cinsiyet=@cinsiyet ", conn);
```

Eğer daha önceden bir kayıt varsa aşağıdaki sorgu çalışır ve veri tabanında güncelleme yapılır.

```
NpgsqlCommand cmd = new NpgsqlCommand("Update
ihtiyac set adet=@Adet Where isim = @isim and
Id=@Id", conn);
```

```
NpgsqlCommand cmd3 = new NpgsqlCommand("Update
ihtiyac set fiyat=@fiyat Where isim = @isim and
Id=@Id", conn);
```

Değilse aşağıdaki sorgu veri tabanına ekleme yapar.


```
NpgsqlCommand cmd = new NpgsqlCommand("insert
into ihtiyac(okul_id,isim,adet,marka,fiyat,tur)" +
"
values(@okul_id,@isim,@adet,@marka,@fiyat,'giysi')",
conn);
```

```
NpgsqlCommand cmd2 = new NpgsqlCommand("insert
into giysi(ihtiyac_id,beden,renk,kumas,cinsiyet)" +
"
values(@ihtiyac_id,@beden,@renk,@kumas,@cinsiyet)"
, conn);
```

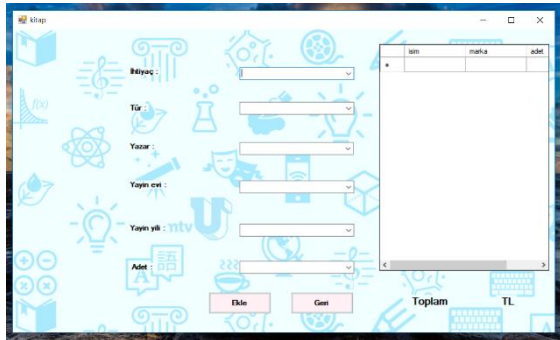
Buradaki sorguda ise combobox'lara kategoriye ait olan ürünler getirilir.

```
komut.CommandText = "SELECT * FROM urun where
kategori = @kategori";
komut.Parameters.AddWithValue("@kategori",
"giysi");
```

Burdaki sorguda ise ekrana okula ait olan ihtiyaçlar getirilir.

```
string kayit = "SELECT isim,marka,adet " +
"from ihtiyac where ihtiyac.okul_id
=@okul_id";
```

Bir önceki sayfaya gelip Kitap ihtiyaçları ekle butonuna bastığımız zaman ise aşağıdaki Kitap sayfası açılır.



Bu sorgu fiyat hesaplamayı sağlar.

```
NpgsqlCommand cmd = new NpgsqlCommand("select
fiyat from urun where ad=@ad and tur = @tur and
yayin_evi = @yayin_evi and yazar = @yazar and
yayin_yili = @yayin_yili", conn);
```

Aşağıdaki sorguda ise veri tabanına ekleme işlemi gerçekleştirilir.

```
NpgsqlCommand kontrol = new NpgsqlCommand("select
adet from ihtiyac where isim=@isim", conn);
```

```
NpgsqlCommand kontrol2 = new
NpgsqlCommand("select ihtiyac_id from kitap where
tur=@tur and yayin_evi=@yayin_evi and yazar=@yazar
and yayin_yili=@yayin_yili", conn);
```

Burada güncelleme işlemi yapılır.

```
NpgsqlCommand cmd = new NpgsqlCommand("Update
ihtiyac set adet=@Adet Where isim = @isim and
Id=@Id", conn);
```

```
NpgsqlCommand cmd3 = new NpgsqlCommand("Update
ihtiyac set fiyat=@fiyat Where isim = @isim and
Id=@Id", conn);
```

Eğer daha önceden ekli değilse ekleme sorgusu çalışır.

```
NpgsqlCommand cmd = new NpgsqlCommand("insert
into ihtiyac(okul_id,isim,adet,marka,fiyat,tur)" +
"
values(@okul_id,@isim,@adet,@marka,10,'kitap')",
conn);
```

```
NpgsqlCommand komut = new NpgsqlCommand("select
max(id) from ihtiyac", conn);
```

```
NpgsqlCommand cmd2 = new NpgsqlCommand("insert
into kitap(ihtiyac_id,tur,yazar,yayin_evi,yayin_yili)" +
"
values(@ihtiyac_id,@tur,@yazar,@yayin_evi,@yayin_yi
li)", conn);
```

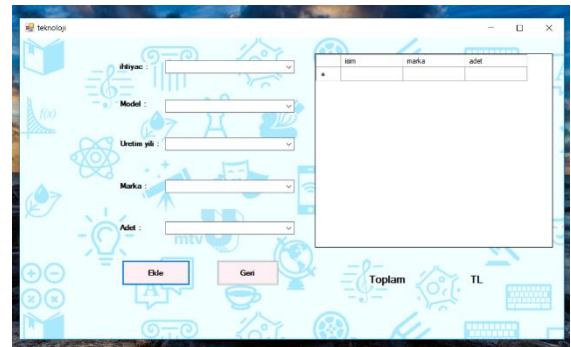
Buradaki sorguda ise combobox'lara kategoriye ait olan ürünler getirilir.

```
komut.CommandText = "SELECT * FROM urun where
kategori = @kategori";
komut.Parameters.AddWithValue("@kategori",
"kitap");
```

Buradaki sorguda ise ekrana okula ait olan ihtiyaçlar getirilir.

```
string kayit = "SELECT isim,marka,adet " +
"from ihtiyac where ihtiyac.okul_id
=@okul_id";
```

Bir sayfa geri gelindikten sonra teknoloji ihtiyacı ekle butonuna tıklandığı zaman aşağıdaki sayfa açılır.



Bu sayfada da diğer ihtiyaç ekleme sayfalarında olduğu gibi aynı işleyiş söz konusudur. Toplam fiyatı hesaplamaya yardımcı olan bir sorgu ve bunun yanında veri tabanına ekleme yapan bir sorgu vardır.

Aşağıdaki sorgu fiyat hesaplamaya yardımcı olan sorgudur.

```
NpgsqlCommand cmd = new NpgsqlCommand("select fiyat from urun where ad=@ad and marka = @marka and model = @model and uretim_yili = @uretim_yili", conn);
```

Buradaki sorguda ise ekrana kategoriye ait olan ürünler getirilir.

```
komut.CommandText = "SELECT * FROM urun where kategori = @kategori";
```

```
komut.Parameters.AddWithValue("@kategori", "teknoloji");
```

Bu sorguda da ekrana okula ait olan ihtiyaçlar getirilir.

```
string kayit = "SELECT isim,marka,adet " + "from ihtiyac where ihtiyac.okul_id = @okul_id";
```

Aşağıdaki sorgu toplam fiyatı hesaplamaya yardımcı olarak kullanılır.

```
string kayit = "SELECT SUM(fiyat) " + "from ihtiyac where ihtiyac.okul_id = @okul_id";
```

Burada teknolojik bir ihtiyaç eklemek istendiğinde çalışan SQL sorguları görülmektedir.

```
NpgsqlCommand kontrol = new NpgsqlCommand("select adet from ihtiyac where isim=@isim and marka=@marka", conn);
```

```
NpgsqlCommand kontrol2 = new NpgsqlCommand("select ihtiyac_id from teknoloji where model=@model and uretim_yili=@uretim_yili ", conn);
```

Daha önceden ekli bir ihtiyaç varsa adet ve toplam fiyatı aşağıdaki sorgu ile güncellenir.

```
NpgsqlCommand cmd = new NpgsqlCommand("Update ihtiyac set adet=@Adet Where isim = @isim and Id=@Id", conn);
```

```
NpgsqlCommand cmd3 = new NpgsqlCommand("Update ihtiyac set fiyat=@fiyat Where isim = @isim and Id=@Id", conn);
```

Eğer daha önceden ekli değilse ekleme sorgusu çalışır.

```
NpgsqlCommand cmd = new NpgsqlCommand("insert into ihtiyac(okul_id,isim,adet,marka,fiyat,tur)" + " values( @okul_id,@isim,@adet,@marka,10,'teknoloji')", conn);
```

Gerekli diğer bilgiler teknoloji tablosuna eklenir.

```
NpgsqlCommand komut = new NpgsqlCommand("select max(id) from ihtiyac", conn); int count = Convert.ToInt32(komut.ExecuteScalar()); NpgsqlCommand cmd2 = new NpgsqlCommand("insert into teknoloji(ihtiyac_id,model,uretim_yili)" +
```

```
" values( @ihtiyac_id,@model,@uretim_yili)", conn);
```

Bir önceki ihtiyaç sayfasında mobilya ihtiyacı ekle butonuna tıklandığında aşağıdaki sayfaya geçiş yapılır.

Bu sayfada mobilya ihtiyaçları eklenir ve diğer ihtiyaçlar toplam fiyatla beraber görüntülenir. Bu sayfada çalışan SQL sorgular aşağıda listelenmiştir.

```
NpgsqlCommand cmd = new NpgsqlCommand("select fiyat from urun where ad=@ad and marka = @marka and renk = @renk and olcu = @olcu", conn);
```

Bu sorgu arka planda toplam fiyatın hesaplanmasında kullanılır.

Bu sorgu ekransa sağ altta gösterilen toplam fiyatın hesaplanmasında kullanılır.

```
string kayit = "SELECT SUM(fiyat) " + "from ihtiyac where ihtiyac.okul_id = @okul_id";
```

Aşağıdaki sorgular ise mobilya ihtiyacını eklemek için ekle butonuna basıldığı zaman çalışmaktadır.

```
NpgsqlCommand kontrol = new NpgsqlCommand("select adet from ihtiyac where isim=@isim and marka=@marka", conn);
```

```
NpgsqlCommand kontrol2 = new NpgsqlCommand("select ihtiyac_id from mobilya where olcu=@olcu and renk=@renk", conn);
```

Burada eğer daha önceden kayıtlı bir ihtiyaç varsa onun güncellenmesi sağlanır.

```
NpgsqlCommand cmd = new NpgsqlCommand("Update ihtiyac set adet=@Adet Where isim = @isim and Id=@Id", conn);
```

```
NpgsqlCommand cmd3 = new NpgsqlCommand("Update ihtiyac set fiyat=@fiyat Where isim = @isim and Id=@Id", conn);
```

Bu ihtiyaç hiç yoksa veri tabanına eklenir.

```
NpgsqlCommand cmd = new NpgsqlCommand("insert into ihtiyac(okul_id,isim,adet,marka,fiyat,tur)" +
```

```

"
values(@okul_id,@isim,@adet,@marka,10,'mobilya')",
conn);

NpgsqlCommand komut = new NpgsqlCommand("select
max(id) from ihtiyac", conn);
NpgsqlCommand cmd2 = new
NpgsqlCommand("insert into
mobilya(ihtiyac_id,renk,olcu)" +
" values(@ihtiyac_id,@renk,@olcu)", conn);

```

Aşağıdaki sorguda ise kayıtlar ekrana getirilir.

```

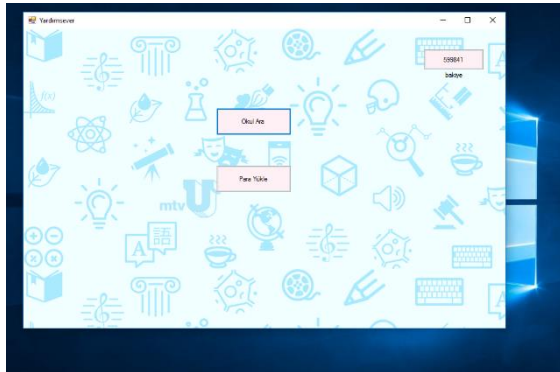
string kayit = "SELECT isim,marka,adet " +
" from ihtiyac where ihtiyac.okul_id
=@okul_id";

NpgsqlCommand komut = new NpgsqlCommand(kayit,
conn);

```

Yardımsever Sayfaları:

Sisteme giriş yaptığınız zaman eyer yetki numaranız 1 ise yardım sever sayfasına yönlendiriliyorsunuz.



Yardım sever ana sayfası yukarıdaki şekilde gösterilmiştir. Bu sayfada 2 adet buton bulunmaktadır. Bunlar Para yükle ve Okul Ara butonlarıdır. Bu butonların dışında sağ üst tarafta yardımseverin bakiyesindeki para miktarı gösterilmektedir.

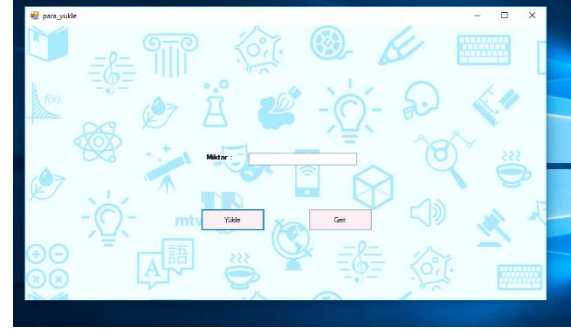
Bakiye miktarı aşağıdaki sorgu ile hesaplanmaktadır.

```

NpgsqlCommand komut = new
NpgsqlCommand("SELECT bakiye from yardımsever
where kAdi=@kullanici_adi", conn);

```

Para yükle butonuna tıklandığı zaman aşağıdaki sayfa açılmaktadır.



Girilen miktar kadar yardımseverin bakiyesine para eklenmektedir. Yükle butonuna basıldığı zaman aşağıdaki SQL sorguları çalışmaktadır.

Yardımseverin bakiyesine girilen miktar eklenir.

```

string kayit1 = "select bakiye from yardımsever where
kAdi=@kullanici_adi ";
NpgsqlCommand komut1 = new
NpgsqlCommand(kayit1, conn);

```

```

komut1.Parameters.AddWithValue("@kullanici_adi",
kullanici_adi);

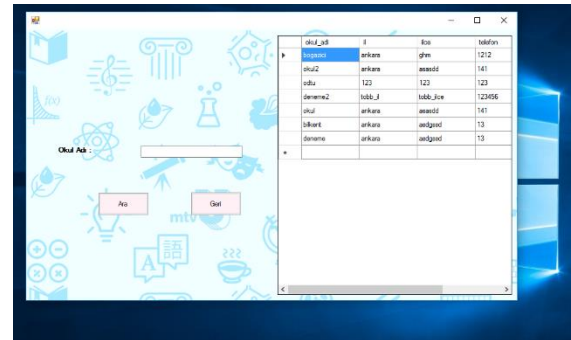
```

```

string kayit = "update yardımsever set
bakiye=@bakiye where kAdi=@kullanici_adi";

```

Yardımsever sayfasındayken Okul Ara butonuna tıklandığında aşağıda görüntülenen sayfaya geçiş yapılır.



Bu sayfada istenilen okul için arama yapılır ve seçilir. Geri butonuna basıldığı zaman bir önceki sayfaya gidilir. Ara butonuna basıldığı zaman ise aranan okulun okul-ihtiyaç sayfasına gidilir.

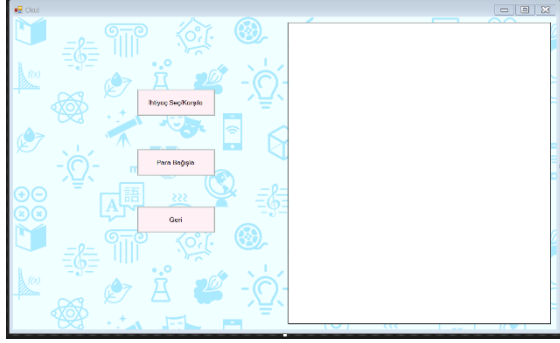
Ara butonuna basıldığı zaman arkada çalışan SQL sorgusu aşağıdaki gibidir.

```

NpgsqlCommand komut = new
NpgsqlCommand("SELECT kullanıcı_adi,il,ilce,telefon
from kullanıcı where yetki =2 and
kullanıcı_adi=@okul_adi ", conn);
komut.Parameters.AddWithValue("@okul_adi",
okul_adi.Text);

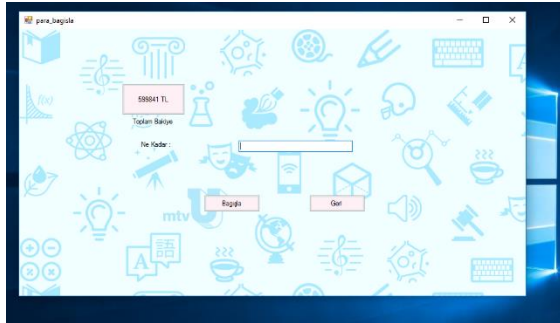
```

Bir sonraki sayfa aşağıda gösterilen Okul-ihitiyaç sayfasıdır.



Bu sayfada ihtiyaç seç ve karşıla, para bağışla ve geri butonları bulunmaktadır. Geri butonu bir önceki sayfa olan okul arama sayfasına yönlendirir. Para bağışla butonu ile seçilmiş olan okula yardımseverin bakiyesinin limiti sınırınca para bağışı yapılabilir.

Para bağışla butonuna tıklandığı zaman Yardım severe ait olan para bağışlama sayfası ekrana gelir.



Bu sayfada yardımsever okula bağış yapabilmektedir. Bağış butonuna tıkladığı zaman bakiyesindeki paranın girilen miktarı kadar düşülüp okulun bakiyesine aktarılmaktadır.

Butona basıldığında aşağıdaki sorgular gerçekleştirilmektedir.

```
NpgsqlCommand komut = new
NpgsqlCommand("SELECT bakiye from yardımsever
where kAdi=@kullanici_adi", conn);
```

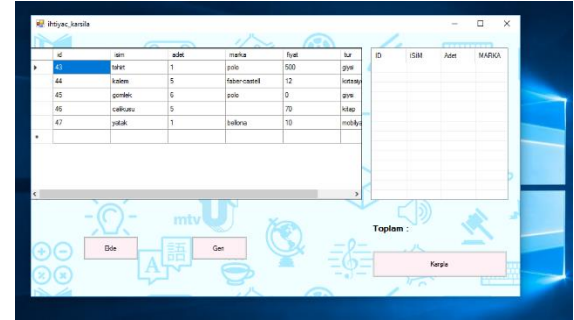
```
komut.Parameters.AddWithValue("@kullanici_adi",
kullanici_adi);
```

```
NpgsqlCommand yardımsever = new
NpgsqlCommand("update yardımsever set bakiye=bakiye
- @ybakiye where kAdi=@kullanici_adi ", conn);
```

```
NpgsqlCommand okul = new NpgsqlCommand("update
okul set bakiye=bakiye + @obakiye where
id=@okul_id", conn);
```

Eğer para bağışla yerine ihtiyaç seç-karşıla butonuna basılırsa yardımseverin ihtiyaç karşıla sayfasına erişim gerçekleştirilir.

Bu tablo aşağıdaki gibidir.



Bu sayfada ekle butonu sağ taraftaki listeye eklenmek istenen eşyaları seçer. Daha sonra sağ altta bulunan Karşıla butonuna tıkladığı zaman listede bulunan elemanlar okul için karşılanmış olur.

Ekle denildiği zaman sistemde çalışan SQL sorgular aşağıdaki gibidir.

```
kayit3 = "update ihtiyac set adet=" + kalanAdet + "
where id = @ihtiyac_id";
kayit4 = "update yardımsever set
bakiye=bakiye-@price where
kullaniciAdi=@kullaniciAdi ";
```

Karşıla butonuna basıldığı zaman sistemde oluşan sorgu aşağıdaki gibidir.

```
string kayit = "insert into
gecmis(okul_id,yardımsever_id,ad,marka,adet,fiyat,tur)
values(@okul_id,1,@ad,@marka,@adet,@fiyat,@tur)";
string kayit2 = "delete from " +
listView1.Items[i].SubItems[5].Text + " where ihtiyac_id
= @ihtiyac_id";
```

```
kayit3 = "update ihtiyac set adet=" + kalanAdet + "
where id = @ihtiyac_id";
kayit4 = "update yardımsever set
bakiye=bakiye-@price where
kullaniciAdi=@kullaniciAdi ";
```

