

SN BEAUTY GÜZELLİK MERKEZİ PROJESİ RAPORU

TARİH :

03/06/2024

AD : SELÇUK

SOYAD : GÜNDÜÇ

PROJE BİLGİLERİ

PROJE ADI : SN BEAUTY

PROJE BİLGİSİ : GÜZELLİK SALONU STOK TAKİP, SATIŞ

VE ÜRÜN LİSTELEME OTOMASYONU

PROJE : C# DİLİ KULLANILARAK YAZILMIŞ

BİR OTOMASYON PROJESİDİR

SELÇUK GÜNDÜÇ

SN BEAUTY GÜZELLİK MERKEZİ PROJESİ RAPORU

GÜZELLİK SALONU OTOMASYONU

SN BEAUTY



shutterstock.com · 2351713871

İÇİNDEKİLER :

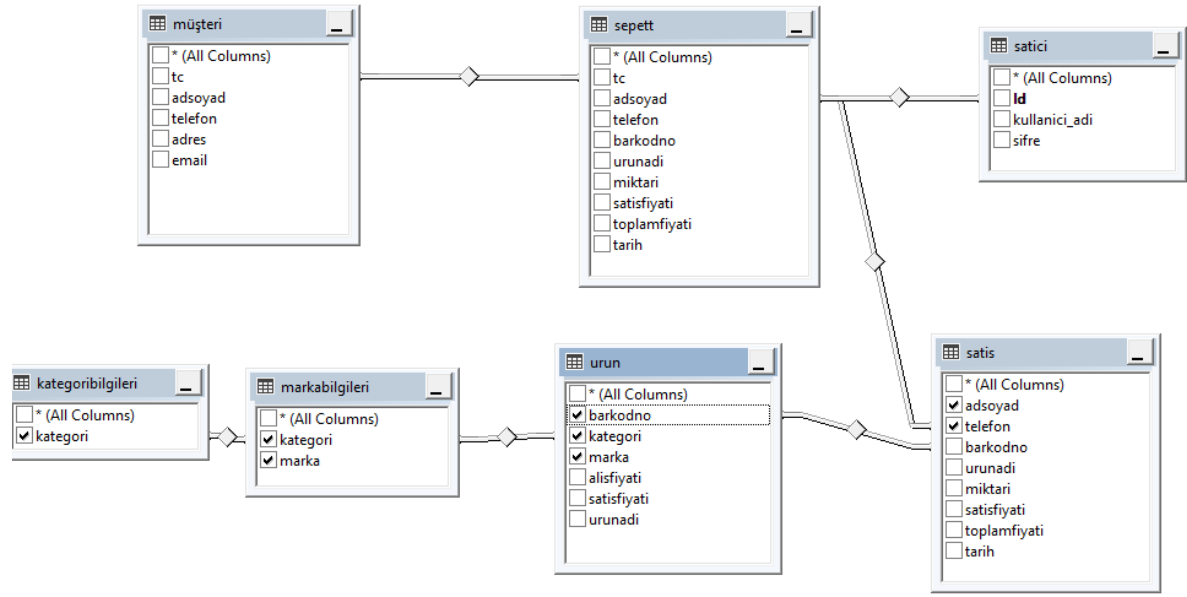
- 1- ÖZET**
- 2- VERİ TABANI DİYAGRAMLARI**
- 3- VERİ TABANI TABLOLAR**
- 4- KULLANICI GİRİŞ SAYFASI**
- 5- KULLANICI ANA SAYDASI**
- 6- SATIŞ PANELİ**
- 7- ŞİFRE YENİLEME SAYFASI**
- 8- KAYIT OLMA SAYFASI**
- 9- ÜRÜN EKLEME SAYFASI**
- 10- ÜRÜN LİSTESİ**
- 11- MÜŞTERİ EKLE**
- 12- MÜŞTERİ LİSTESİ**
- 13- KATEGORİ SAYFASI**
- 14- MARKA SAYFASI**
- 15- SATIŞ LİSTESİ**
- 16- KURU CİTLER**
- 17- NORMAL CİTLER**

ÖZET:

C# dilini kullanarak cilt bakım ürünlerinin satışını yapan,kullanıcı girişinin ve satıcı girişinin yapıldığı e-ticaret sitesi yaptım. Visual Studio ve SQL Server programlarını kullandım.

Satıcı sayfasında ürün ekleme kategori marka oluşturma sayfaları yaptım ve birbirine bağladım. Ürünler eklendikten sonra ürün listeleme sayfasında detaylıca görüp güncelleme silme yapabilir. Aynı şekilde müşteri ekleyip listesine ulaşabiliyor. Ve satıcı sayfasının ilk girişine sepet oluşturdum sepette girdiği miktara göre fiyatı verip satışını yapmakta.

VERİ TABANI DİYAGRAMLARI



SATICI PANELİ MÜŞTERİ KAYITLARI

tc	adsoyad	telefon	adres	email	KullaniciAdi	Sifre
16625348899	selçuk çetin...	5541633165	sancaktepe	selcukgnd@...	NULL	NULL
58585858	sedat	5857545758	sultanbeyli	selma@gm...	NULL	NULL
741852963	eda	78451548	pendik	eda@gmail...	NULL	NULL
987654	emre arık	123456	pendik	emrenos@g...	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

MÜŞTERİ LİSTESİ

Id	kullanici_a...	sifre
1	federal	6534
2	selcuk	6598
NULL	NULL	NULL

SATICI LİSTESİ

Id	kullanici_adi	sifre
1	federal	6534
NULL	NULL	NULL

ÜRÜN LİSTESİ

alisfiyati	satisfiyati	urunadi
36,50	58,00	kırmızı oje
20,00	40,00	cilt nemlen...
40,00	80,00	batman ma...
65,00	70,00	siyah nokta ...
NULL	NULL	NULL

MARKA KATEGORİ BİLGİLERİ

kategori	marka
nemlendirici	nivea
nemlendirici	garnier
nemlendirici	the purest
yüz bakım	nivea
kozmetik	minevital
maske	garnier yüz ...
el kremi	arko
sivilce	sinoz
NULL	NULL


SEPET BİLGİLERİ

[illegible]

SATIŞ LİSTESİ

[illegible]

KULLANICI GİRİŞ SAYFASI

*SN BEAUTY*

KULLANICI GİRİŞİ

Kullanıcı Adı:

Şifre:

GİRİŞ

[Kayıt Ol](#) [Şifremi Unuttum](#)

SATICI GİRİŞİ

Kullanıcı Adı:

Şifre:

GİRİŞ

[Kayıt Ol](#) [Şifremi Unuttum](#)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.Sql;
using System.Data.SqlClient;

namespace SNbeauty
{
    4 başvuru
    public partial class kullanıcı_giris : Form
    {
        // Static değişkenler kullanıcı bilgilerini saklamak için kullanılır
        public static string sid, ssifre;
        public static string mid, msifre;
        public static string zkid, zsifre;

        1 başvuru
        public kullanıcı_giris()
        {
            InitializeComponent(); // Form bileşenlerini başlatır
        }

        // Veritabanı bağlantı dizesi
        static string constring = "Data Source=FEDERAL\\SQLEXPRESS01;Initial Catalog=müşterilistes;Integrated Security=True";
        SqlConnection baglanti = new SqlConnection(constring); // Veritabanı bağlantısını oluşturur
    }
}
```



```

0 başvuru
private void pictureBox1_Click(object sender, EventArgs e)
{
}

1 başvuru
private void button1_Click(object sender, EventArgs e)
{
    // Kullanıcı adı ve şifre alanları boş değilse
    if (textBox1.Text != "" && textBox2.Text != "")
    {
        try
        {
            // Eğer bağlantı açıksa kapatır
            if (baglanti.State == ConnectionState.Open)
            {
                baglanti.Close();
            }
            // Bağlantıyı açar
            baglanti.Open();

            // Yönetici tablosundan kullanıcı adı ve şifre kontrolü yapar
            SqlCommand kullanici = new SqlCommand("Select yönetici.kullanici_adi, yönetici.sifre FROM yönetici WHERE kullanici_adi = @kadi and sifre = @ksifre", baglanti);
            kullanici.Parameters.AddWithValue("@kadi", SqlDbType.NVarChar).Value = textBox1.Text;
            kullanici.Parameters.AddWithValue("@ksifre", SqlDbType.Int).Value = textBox2.Text;
            SqlDataReader oku = kullanici.ExecuteReader();

            // Eğer kullanıcı bilgileri doğruysa
            if (oku.Read())
            {
                zkid = textBox1.Text; // Kullanıcı adını saklar
                MessageBox.Show("Giriş Başarılı, HOŞGELDİN GÜZELLİK");
                kullanici_anasayfa yeni = new kullanici_anasayfa(); // Ana sayfa formunu açar
                this.Hide(); // Mevcut formu gizler
                yeni.Show(); // Yeni formu gösterir
            }
            else
            {
                // Eğer kullanıcı bilgileri yanlışsa
                MessageBox.Show("Böyle bir kayıt yok.");
            }
            baglanti.Close(); // Bağlantıyı kapatır
        }
        catch
        {
            // Bir hata oluşursa mesaj gösterir
            MessageBox.Show("Hata!!");
        }
    }
    else
    {
        // Kullanıcı adı veya şifre alanı boşsa mesaj gösterir
        MessageBox.Show("Boş alanları doldurun.");
    }
}

1 başvuru
private void groupBox2_Enter(object sender, EventArgs e)
{
}

1 başvuru
private void linkLabel1_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    // Şifre yenileme formunu açar
    sifre_yenile Sifre_Yenile = new sifre_yenile();
    Sifre_Yenile.Show();
}

```

```

1 başvuru
private void groupBox1_Enter(object sender, EventArgs e)
{
}

1 başvuru
private void linkLabel3_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    // Form6'yı açar
    Form6 form6 = new Form6();
    form6.Show();
}

1 başvuru
private void panel1_Paint(object sender, PaintEventArgs e)
{
}

1 başvuru
private void pictureBox1_Click_1(object sender, EventArgs e)
{
}

1 başvuru
private void linkLabel2_LinkClicked(object sender, LinkLabelLinkClickedEventArgs e)
{
    // Kullanıcı kayıt formunu açar
    kullanıcı_kayit kullanıcı_Kayit = new kullanıcı_kayit();
    kullanıcı_Kayit.Show();
}

// Satıcı girişi: kullanıcı adı ve şifre alanları boş değilse
if (textBox4.Text != "" && textBox3.Text != "")
{
    try
    {
        // Eğer bağlantı açıksa kapatır
        if (baglanti.State == ConnectionState.Open)
            baglanti.Close();
        // Bağlantıyı açar
        baglanti.Open();

        // Satıcı tablosundan kullanıcı adı ve şifre kontrolü yapar
        SqlCommand kullanici = new SqlCommand("Select satıcı.kullanıcı_adi, satıcı.sifre FROM satıcı WHERE kullanıcı_adi = @kadi and sifre = @ksifre", baglanti);
        kullanici.Parameters.AddWithValue("@kadi", SqlDbType.NVarChar).Value = textBox4.Text;
        kullanici.Parameters.AddWithValue("@ksifre", SqlDbType.Int).Value = textBox3.Text;
        SqlDataReader oku = kullanici.ExecuteReader();

        // Eğer kullanıcı bilgileri doğruysa
        if (oku.Read())
        {
            {
                zkid = textBox4.Text; // Kullanıcı adını saklar
                MessageBox.Show("Giriş Başarılı, HOŞGELDİN GÜZELLİK");
                satıcı_anasayfa yeni = new satıcı_anasayfa(); // Ana sayfa formunu açar
                this.Hide(); // Mevcut formu gizler
                yeni.Show(); // Yeni formu gösterir
            }
            else
            {
                // Eğer kullanıcı bilgileri yanlışsa
                MessageBox.Show("Böyle bir kayıt yok.");
            }
            baglanti.Close(); // Bağlantıyı kapatır
        }
        catch
        {
            // Bir hata oluşursa mesaj gösterir
            MessageBox.Show("Hata!!");
        }
    }
    else
    {
        // Kullanıcı adı veya şifre alanı boşsa mesaj gösterir
        MessageBox.Show("Boş alanları doldurun.");
    }
}

1 başvuru
private void kullanıcı_giris_Load(object sender, EventArgs e)

```

Kullanıcı adı ve şifre girildiğinde giriş butonuna tıklayarak kullanıcı ana sayfaya ya da satıcı paneline giriş yapılabilir. Burda Sql e bağlayarak oluşturduğum kişi kayıt sayfasındaki doğrulamaya göre buradan giriş yapılabilmektedir. Giriş yapıldığında ya da hata olduğunda validasyonlarla destekledim.

KULLANICI ANA SAYFA



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace SNbeauty
{
    5 başvuru
    public partial class kullanıcı_anasayfa : Form
    {
        // kullanıcı_anasayfa formunun yapıcı metodu
        1 başvuru
        public kullanıcı_anasayfa()
        {
            InitializeComponent(); // Form bileşenlerini başlatır
        }

        0 başvuru
        private void numericUpDown1_ValueChanged(object sender, EventArgs e)
        {
        }

        1 başvuru
        private void button4_Click(object sender, EventArgs e)
        {
            // Akneli ciltler için yeni formu açar
            akneliCiltler Cilt = new akneliCiltler();
            Cilt.Show();
        }

        1 başvuru
        private void button5_Click(object sender, EventArgs e)
        {
            // Karma ciltler için yeni formu açar
            karmaCiltler Cilt = new karmaCiltler();
            Cilt.Show();
        }

        1 başvuru
        private void Form2_Load(object sender, EventArgs e)
        {
        }

        1 başvuru
        private void label6_Click(object sender, EventArgs e)
        {
        }
    }
}
```

7 tane form sayfası açıp bu ana sayfadaki butonlara bağladım. Herhangi bir tanesi seçildiğinde o sayfaya giriş yapılacaktır.

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN BEAUTY

SN

Satıcı Paneli

Müşteri İşlemleri

TC

Ad Soyad

Telefon

Müşteri İşlemleri

Barkod No

Ürün Adı

Miktar

Satış Fiyatı

Toplam Fiyat

Ürün Ekle

Müşteri Ekle

Ürün Listesi

Müşteri Listesi

Kategori

Marka

Sil

Ekle

Genel Toplam

Satış Yap

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlClient;

namespace SNbeauty
{
    5 bağırvuru
    public partial class satıcı_anasayfa : Form
    {
        // Formun yapıcı metodu
        1 bağırvuru
        public satıcı_anasayfa()
        {
            InitializeComponent(); // Form bileşenlerini başlatır
        }

        // Veritabanı bağlantısı oluşturuluyor
        SqlConnection bağlantı = new SqlConnection("Data Source=FEDERAL\\SQLEXPRESS01;Initial Catalog=müşteriListes;Integrated Security=True");

        // DataSet nesnesi tanımlanıyor
        DataSet daset = new DataSet();

        // Sepeti listeleme metod
        4 bağırvuru
        private void sepetlistele()
        {
            bağlantı.Open(); // Veritabanı bağlantısı açılıyor
            SqlDataAdapter adtr = new SqlDataAdapter("select *from sepett", bağlantı); // Sepet tablosundan verileri çeker
            adtr.Fill(daset, "sepett"); // Verileri DataSet'e doldurur
            dataGridView1.DataSource = daset.Tables["sepett"]; // DataGridView'e verileri atar
            bağlantı.Close(); // Veritabanı bağlantısını kapatır
        }

        // Barkod kontrolü için durum değişkeni
        private bool durum;

        // Barkod kontrol metodu
        0 bağırvuru
        private void barkodkontrol()
        {
            durum = true; // Durumu true olarak ayarlar
            bağlantı.Open(); // Veritabanı bağlantısını açar
            SqlCommand komut = new SqlCommand("select *from sepett", bağlantı); // Sepet tablosundan verileri çeker
            SqlDataReader read = komut.ExecuteReader(); // Veri okuyucu nesnesi
```

```

// Ürün ekleme butonu tıklama olayı
1 başvuru
private void button1_Click(object sender, EventArgs e)
{
    urunEkle Ekle = new urunEkle(); // Yeni ürün ekleme formunu oluşturur
    Ekle.Show(); // Formu gösterir
}

// Form yüklendiğinde sepeti listeleyen metod çağrılır
1 başvuru
private void Form5_Load(object sender, EventArgs e)
{
    sepetListele();
}

// Genel toplamı hesaplayan metod
3 başvuru
private void hesapla()
{
    try
    {
        baglanti.Open(); // Veritabanı bağlantısını açar
        SqlCommand komut = new SqlCommand("select sum(toplamfiyati) from sepet", baglanti); // Sepet tablosundaki toplam fiyatları toplar
        lblGenelToplam.Text = komut.ExecuteScalar() + " TL "; // Sonucu etiket üzerinde gösterir
        baglanti.Close(); // Veritabanı bağlantısını kapatır
    }
    catch (Exception)
    {
        // Hata durumunda yapılacak işlemler
    }
}

// Ürün listesi butonu tıklama olayı
1 başvuru
private void button3_Click(object sender, EventArgs e)
{
    urunListesi Liste = new urunListesi(); // Ürün listesi formunu oluşturur
    Liste.Show(); // Formu gösterir
}

// Müşteri listele butonu tıklama olayı
1 başvuru
private void button2_Click(object sender, EventArgs e)
{
    frmMusteriListele liste = new frmMusteriListele(); // Müşteri listeleme formunu oluşturur
    liste.ShowDialog(); // Formu diyalog olarak gösterir
}

// Kategori butonu tıklama olayı
1 başvuru
private void button8_Click(object sender, EventArgs e)
{
    frmkategori kategori = new frmkategori(); // Kategori formunu oluşturur
    kategori.ShowDialog(); // Formu diyalog olarak gösterir
}

// Marka butonu tıklama olayı
1 başvuru
private void button7_Click(object sender, EventArgs e)
{
    frmMarka marka = new frmMarka(); // Marka formunu oluşturur
    marka.ShowDialog(); // Formu diyalog olarak gösterir
}

// TC numarası metin kutusu değiştiğinde gerçekleşecek olay
1 başvuru
private void txtTc_TextChanged(object sender, EventArgs e)
{
    if (txtTc.Text == "")
    {
        txtAdSoyad.Text = ""; // Ad soyad alanını temizler
        txtTelefon.Text = ""; // Telefon alanını temizler
    }
    baglanti.Open(); // Veritabanı bağlantısını açar
    SqlCommand komut = new SqlCommand("select *from müşteri where tc like '" + txtTc.Text + "'", baglanti); // TC numarasına göre müşteri bilgilerini çeker
    SqlDataReader read = komut.ExecuteReader(); // Veri okuyucu nesnesi
    while (read.Read())
    {
        txtAdSoyad.Text = read["adsoyad"].ToString(); // Ad soyadı alanına yazar
        txtTelefon.Text = read["telefon"].ToString(); // Telefon alanına yazar
    }
    baglanti.Close(); // Veritabanı bağlantısını kapatır
}

// Barkod numarası metin kutusu değiştiğinde gerçekleşecek olay
1 başvuru
private void txtBarkodNo_TextChanged(object sender, EventArgs e)
{
    temizle(); // Metin kutularını temizler
    baglanti.Open(); // Veritabanı bağlantısını açar
    SqlCommand komut = new SqlCommand("select *from urun where barkodno like '" + txtBarkodNo.Text + "'", baglanti); // Barkod numarasına göre ürün bilgilerini çeker
    SqlDataReader read = komut.ExecuteReader(); // Veri okuyucu nesnesi
    while (read.Read())
    {
        txtÜrünAdı.Text = read["urunadi"].ToString(); // Ürün adını metin kutusuna yazar
        txtSatışFiyatı.Text = read["satışfiyatı"].ToString(); // Satış fiyatını metin kutusuna yazar
    }
    baglanti.Close(); // Veritabanı bağlantısını kapatır
}

// Metin kutularını temizleyen metod
2 başvuru
private void temizle()
{
    if (txtBarkodNo.Text == "")
    {
        foreach (Control item in groupBox2.Controls)
        {
            if (item is TextBox)
            {
                if (item != txtMiktarı)
                {
                    item.Text = ""; // Metin kutularını temizler
                }
            }
        }
    }
}

// Ürün ekleme butonu tıklama olayı
1 başvuru
private void btnEkle_Click(object sender, EventArgs e)
{
    baglanti.Open(); // Veritabanı bağlantısını açar
    SqlCommand komut = new SqlCommand("insert into sepet(tc, adsoyad, telefon, barkodno, urunadi, miktarı, satışfiyatı, toplamfiyatı, tarih) values(@tc, @adsoyad, @telefon, @barkodno, @urunadi, "
    komut.Parameters.AddWithValue("@tc", txtTc.Text); // Parametreleri ekler
    komut.Parameters.AddWithValue("@adsoyad", txtAdSoyad.Text);
    komut.Parameters.AddWithValue("@telefon", txtTelefon.Text);
    komut.Parameters.AddWithValue("@barkodno", txtBarkodNo.Text);
    komut.Parameters.AddWithValue("@urunadi", txtÜrünAdı.Text);
    komut.Parameters.AddWithValue("@miktarı", int.Parse(txtMiktarı.Text));
    komut.Parameters.AddWithValue("@satışfiyatı", double.Parse(txtSatışFiyatı.Text));
    komut.Parameters.AddWithValue("@toplamfiyatı", double.Parse(txtToplamFiyatı.Text));
    komut.Parameters.AddWithValue("@tarih", DateTime.Now.ToString());
    komut.ExecuteNonQuery(); // Sorguyu çalıştırır
    baglanti.Close(); // Veritabanı bağlantısını kapatır
    dataSet.Tables["sepet"].Clear(); // DataSet'i temizler
    sepetListele(); // Sepeti listele
    temizle(); // Metin kutularını temizle
    hesapla(); // Genel toplamı hesapla

    foreach (Control item in groupBox2.Controls)
    {
        if (item is TextBox)
        {
            if (item != txtMiktarı)
            {
                item.Text = ""; // Metin kutularını temizler
            }
        }
    }
}

```

```

// Miktar metin kutusu değiştiğinde gerçekleşecek olay
1bayyuru
private void txtMiktarı_TextChanged(object sender, EventArgs e)
{
    try
    {
        txtToplamFiyat.Text = (double.Parse(txtMiktarı.Text) * double.Parse(txtSatışFiyatı.Text)).ToString(); // Toplam fiyatı hesaplar
    }
    catch (Exception)
    {
        // Hata durumunda yapılacak işlemler
    }
}

// Satış fiyatı metin kutusu değiştiğinde gerçekleşecek olay
1bayyuru
private void txtSatışFiyatı_TextChanged(object sender, EventArgs e)
{
    try
    {
        txtToplamFiyat.Text = (double.Parse(txtMiktarı.Text) * double.Parse(txtSatışFiyatı.Text)).ToString(); // Toplam fiyatı hesaplar
    }
    catch (Exception)
    {
        // Hata durumunda yapılacak işlemler
    }
}

// Sil butonu tıklama olayı
1bayyuru
private void btnSil_Click(object sender, EventArgs e)
{
    baglanti.Open(); // Veritabanı bağlantısını açar
    SqlCommand komut = new SqlCommand("delete from sepett where barkodno=" + dataGridView1.CurrentRow.Cells["barkodno"].Value.ToString() + " ", baglanti); // Sepetten ürünü siler
    komut.ExecuteNonQuery(); // Sorguyu çalıştırır
    baglanti.Close(); // Veritabanı bağlantısını kapatır
    MessageBox.Show("Ürün sepetten çıkarıldı"); // Bilgilendirme mesajı
    daset.Tables["sepett"].Clear(); // DataSet'i temizler
    sepetlistele(); // Sepeti listele
    hesapla(); // Genel toplamı hesapla
}

// Satış listesi butonu tıklama olayı
0bayyuru
private void btnSatışListesi_Click(object sender, EventArgs e)
{
    frmSatışListele satis = new frmSatışListele(); // Satış listeleme formunu oluşturur
    satis.ShowDialog(); // Formu diyalog olarak gösterir
}

// Satış yap butonu tıklama olayı
1bayyuru
private void btnSatışYap_Click(object sender, EventArgs e)
{
    for (int i = 0; i < dataGridView1.Rows.Count - 1; i++)
    {
        baglanti.Open(); // Veritabanı bağlantısını açar
        SqlCommand komut = new SqlCommand("insert into satis(adsoyad, telefon, barkodno, urunadi, miktar, satisfiyati, toplamfiyati, tarih, tc) values(@adsoyad, @telefon, @barkodno, @urunadi, @miktar, @satisfiyati, @toplamfiyati, @tarih, @tc)", baglanti);
        komut.Parameters.AddWithValueValue("@adsoyad", txtAdSoyad.Text); // Parametreleri ekler
        komut.Parameters.AddWithValueValue("@telefon", txtTelefon.Text);
        komut.Parameters.AddWithValueValue("@barkodno", dataGridView1.Rows[i].Cells["barkodno"].Value.ToString());
        komut.Parameters.AddWithValueValue("@urunadi", dataGridView1.Rows[i].Cells["urunadi"].Value.ToString());
        komut.Parameters.AddWithValueValue("@miktar", int.Parse(dataGridView1.Rows[i].Cells["miktar"].Value.ToString()));
        komut.Parameters.AddWithValueValue("@satisfiyati", double.Parse(dataGridView1.Rows[i].Cells["satisfiyati"].Value.ToString()));
        komut.Parameters.AddWithValueValue("@toplamfiyati", double.Parse(dataGridView1.Rows[i].Cells["toplamfiyati"].Value.ToString()));
        komut.Parameters.AddWithValueValue("@tarih", DateTime.Now.ToString());
        komut.Parameters.AddWithValueValue("@tc", txtTc.Text);
        komut.ExecuteNonQuery(); // Sorguyu çalıştırır
        baglanti.Close(); // Veritabanı bağlantısını kapatır
    }
    baglanti.Open(); // Veritabanı bağlantısını açar
    SqlCommand komut3 = new SqlCommand("delete from sepett ", baglanti); // Sepeti temizler
    komut3.ExecuteNonQuery(); // Sorguyu çalıştırır
    baglanti.Close(); // Veritabanı bağlantısını kapatır
    daset.Tables["sepett"].Clear(); // DataSet'i temizler
    sepetlistele(); // Sepeti listele
    hesapla(); // Genel toplamı hesapla
}

```

Sayfanın üst tarafında 7 tane seçenek bulunmaktadır. Kod bloklarında da sayfaların formlarını Show.dialog() ile birbirine bağladım. Ortada bulunan müşterilerin sepetleri ve genel toplamı veren hesapla() komutunu ekle butonuna ve sil butonuna da ekledim. barkodkontrol(); da yaptığım şey ise aynı barkod numaralı ürünleri alt alta eklemeyi miktar olarak arttır. Bilgileri girerken miktarı arttırdığınızda ise yine aynı sütunda verir. Temizle(); de ise textboxlara girilen bilgileri sonradan temizlemek için kullandım.

ŞİFRE YENİLEME

ŞİFRE YENİLEME

Kullanıcı Adı:

Yeni Şifre:

KAYDET

```
public partial class sifre_yenile : Form
{
    // Form yapıcısı
    1 bayyuru
    public sifre_yenile()
    {
        InitializeComponent(); // Form bileşenlerini başlatır
    }

    1 bayyuru
    private void label2_Click(object sender, EventArgs e)
    {
    }

    1 bayyuru
    private void sifre_yenile_Load(object sender, EventArgs e)
    {
    }

    // Şifre yenileme butonu tıklama olayı
    1 bayyuru
    private void button1_Click(object sender, EventArgs e)
    {
        // Kullanıcı adı ve yeni şifreyi alır
        string kullanic Adi = textBox1.Text;
        string yeniSifre = textBox2.Text;

        // Veritabanı bağlantı dizesi
        string connectionString = "Data Source = FEDERAL\\SQLEXPRESS01; Initial Catalog = musterilistes; Integrated Security = True";

        try
        {
            // Veritabanı bağlantısını açar ve kullanır
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                connection.Open(); // Veritabanı bağlantısını açar

                // Kullanıcı adıyla eşleşen kaydın şifresini güncelleyen SQL sorgusu
                string updateQuery = "UPDATE MusteriKaydı SET Sifre = @YeniSifre WHERE KullaniciAdi = @KullaniciAdi";

                // SQL komutu oluşturur ve parametreleri ekler
                using (SqlCommand command = new SqlCommand(updateQuery, connection))
                {
                    command.Parameters.Add("@YeniSifre", SqlDbType.VarChar).Value = yeniSifre; // Yeni şifre parametresi
                    command.Parameters.Add("@KullaniciAdi", SqlDbType.VarChar).Value = kullanic Adi; // Kullanıcı adı parametresi
                }
            }
        }
    }
}
```

```

        command.Parameters.Add("@YeniSifre", SqlDbType.VarChar).Value = yeniSifre; // Yeni şifre parametresi
        command.Parameters.Add("@KullaniciAdi", SqlDbType.VarChar).Value = kullaniciAdi; // Kullanıcı adı parametresi

        // Komutu çalıştırır ve etkilenen satır sayısını alır
        int rowsAffected = command.ExecuteNonQuery();
        if (rowsAffected > 0)
        {
            // Güncelleme başarılıysa mesaj gösterir
            MessageBox.Show("Şifre başarıyla güncellendi.");
        }
        else
        {
            // Kullanıcı bulunamazsa veya başka bir hata oluşursa mesaj gösterir
            MessageBox.Show("Kullanıcı bulunamadı veya bir hata oluştu.");
        }
    }
}

catch (Exception ex)
{
    // Hata oluşursa mesaj gösterir
    MessageBox.Show("Bir hata oluştu: " + ex.Message);
}
}

1 başvuru
private void label1_Click(object sender, EventArgs e)
{
}

1 başvuru
private void textBox1_TextChanged(object sender, EventArgs e)
{
}

1 başvuru
private void label23_Click(object sender, EventArgs e)
{
}
}
}

```

Kullanıcı adını ve yeni şifreyi girdikten sonra veri tabanındaki musterikaydı listesinde şifre güncellenecek.

KULLANICI KAYIT

KULLANICI KAYIT

Kullanıcı Adı:

Şifre:

KAYDET

```
{
    // Veritabanı bağlantı dizesi
    static string constring = "Data Source=FEDERAL\\SQLEXPRESS01;Initial Catalog=MüşteriListes;Integrated Security=True";
    SqlConnection baglanti = new SqlConnection(constring); // SQL bağlantı nesnesi oluşturulur

    // Form yapıcı metodu
    1 bagyuru
    public kullanici_kayit()
    {
        InitializeComponent(); // Form bileşenlerini başlatır
    }

    1 bagyuru
    private void Form4_Load(object sender, EventArgs e)
    {
    }

    1 bagyuru
    private void button1_Click(object sender, EventArgs e)
    {
        try
        {
            // Kullanıcı adı ve şifre alanlarının boş olmadığını kontrol etme
            if (string.IsNullOrEmpty(textBox4.Text) || string.IsNullOrEmpty(textBox3.Text))
            {
                MessageBox.Show("Kullanıcı adı ve şifre alanları boş olamaz.");
                return; // İşlemi sonlandırır
            }

            if (baglanti.State == ConnectionState.Open)
                baglanti.Close(); // Bağlantı açıksa kapatır

            baglanti.Open(); // Veritabanı bağlantısını açar

            // SQL sorgusu
            string insertQuery = "INSERT INTO yönetici (kullanici__adi, sifre) VALUES (@kad, @ksifre)";
            SqlCommand kayit = new SqlCommand(insertQuery, baglanti); // Komut oluşturur

            // Parametre ekleme
            kayit.Parameters.AddWithValue("@kad", textBox4.Text); // Kullanıcı adı parametresi
            kayit.Parameters.AddWithValue("@ksifre", textBox3.Text); // Şifre parametresi

            // Sorguyu çalıştırma
            int rowsAffected = kayit.ExecuteNonQuery(); // Sorguyu veritabanında çalıştırır

            // Bağlantıyı kapatma
            baglanti.Close();

            if (rowsAffected > 0)
            {
                MessageBox.Show("Başarıyla eklendi."); // Başarılı ekleme mesajı
            }
        }
    }
}
```

Sql ile bağlantılı bir şekilde kayıt eder ve giriş için önemli bir sayfadır.

ÜRÜN EKLEME SAYFASI

Yeni Ürün

Barkod Kodu

Kategori

Marka

Alış Fiyatı

Satış Fiyatı

Ürün Adı:

ekle

```
1 bagvuru
public partial class urunEkle : Form
{
    1 bagvuru
    public urunEkle()
    {
        InitializeComponent(); // Form bileşenlerini başlatır
    }

    // SQL bağlantısı oluşturma
    SqlConnection baglanti = new SqlConnection("Data Source=FEDERAL\\SQLEXPRESS01;Initial Catalog=musteriIistes;Integrated Security=True");

    // Kategori verilerini ComboBox'a yükleme
    1 bagvuru
    private void kategorigeticir()
    {
        baglanti.Open(); // Bağlantıyı aç
        SqlCommand komut = new SqlCommand("select *from kategori", baglanti); // SQL sorgusu oluşturun
        SqlDataReader read = komut.ExecuteReader(); // Verileri oku
        while (read.Read()) // Tüm verileri okuyana kadar
        {
            combokategori.Items.Add(read["kategori"].ToString()); // ComboBox'a kategorileri ekle
        }
        baglanti.Close(); // Bağlantıyı kapat
    }

    // Kategori ComboBox'ı değiştirildiğinde çalışan olay
    1 bagvuru
    private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
    {
        comboMarka.Items.Clear(); // Marka ComboBox'ını temizle
        comboMarka.Text = ""; // Marka ComboBox'ını sıfırla
        baglanti.Open(); // Bağlantıyı aç
        SqlCommand komut = new SqlCommand("select kategori,marka from markabilgileri where kategori='" + comboBox1.SelectedItem + "' ", baglanti); // SQL sorgusu
        SqlDataReader read = komut.ExecuteReader(); // Verileri oku
        while (read.Read()) // Tüm verileri okuyana kadar
        {
            comboMarka.Items.Add(read["marka"].ToString()); // ComboBox'a markaları ekle
        }
        baglanti.Close(); // Bağlantıyı kapat
    }

    1 bagvuru
    private void btnyeniekle_Click(object sender, EventArgs e)
    {
        baglanti.Open(); // Bağlantıyı aç
        SqlCommand komut = new SqlCommand("insert into urun(barkodno,kategori,marka,alisfiyati,satisfiyati,urunadi) values(@barkodno,@kategori,@marka,@alisfiyati,@satisfiyati,@urunadi)", baglanti);
        komut.Parameters.AddWithValue("@barkodno", txtBarkodNo.Text); // Parametrelerle değerleri ekle
        komut.Parameters.AddWithValue("@kategori", combokategori.Text);
        komut.Parameters.AddWithValue("@marka", comboMarka.Text);
        komut.Parameters.AddWithValue("@alisfiyati", double.Parse(txtAlisfiyati.Text));
        komut.Parameters.AddWithValue("@satisfiyati", double.Parse(txtSatisfiyati.Text));
    }
}
```

```

        komut.Parameters.AddWithValue("@marka", comboMarka.Text);
        komut.Parameters.AddWithValue("@alisfiyati", double.Parse(txtAlisfiyatı.Text));
        komut.Parameters.AddWithValue("@satisfiyati", double.Parse(txtSatısFiyatı.Text));
        komut.Parameters.AddWithValue("@urunadi", txtÜrünAdı.Text);
        komut.ExecuteNonQuery(); // Sorguyu çalıştır
        baglanti.Close(); // Bağlantıyı kapat
        MessageBox.Show("Ürün Eklendi!"); // Başarı mesajı
        foreach (Control item in groupBox1.Controls) // Grup kutusu içindeki tüm kontrol elemanlarını döngüye al
        {
            if (item is TextBox) // Eğer kontrol bir TextBox ise
            {
                item.Text = ""; // TextBox içeriğini temizle
            }
            if (item is ComboBox) // Eğer kontrol bir ComboBox ise
            {
                item.Text = ""; // ComboBox seçimini temizle
            }
        }
    }

    // Form yüklendiğinde çalışan olay
    1 başvuru
    private void ürünEkle_Load(object sender, EventArgs e)
    {
        kategorigetir(); // Kategorileri yükleme metodu çağrılır
    }

    // Barkod numarası TextBox'ı değiştiğinde çalışan olay (boş, kullanılmıyor)
    0 başvuru
    private void BarkodNotxt_TextChanged(object sender, EventArgs e)
    {
    }

    0 başvuru
    private void groupBox2_Enter(object sender, EventArgs e)
    {
    }

    1 başvuru
    private void groupBox1_Enter(object sender, EventArgs e)
    {
    }

    0 başvuru
    private void label13_Click(object sender, EventArgs e)
    {
    }
}

```

Burada kategori ve marka form sayfasında oluşturulan bilgileri görebilirsiniz. Girdiğiniz bilgilerin hepsini ürün listesine ekler.

ÜRÜN LİSTESİ

Barkod Koduna göre ara

Var Olan Ürün

Barkod Kodu

Kategori

Marka

Alış Fiyatı

Satış Fiyatı

Ürün Adı:

Güncelle

Sil

Kategori

Marka

marka güncelle

```
private void btnvarolanaekle_Click(object sender, EventArgs e)
{
    baglanti.Open(); // Veritabanı bağlantısını aç

    // UPDATE SQL komutu ile urun tablosundaki belirli bir ürünün bilgilerini güncelliyoruz.
    SqlCommand komut = new SqlCommand("update urun set alisfiyati=@alisfiyati, satisfiyati=@satisfiyati, urunadi=@urunadi where barkodno=@barkodno", baglanti);

    // Parametrelerimize degerleri atiyoruz
    komut.Parameters.AddWithValue("@barkodno", BarkodNotxt.Text); // Barkod numarası
    komut.Parameters.AddWithValue("@alisfiyati", double.Parse(AlisFiyatitxt.Text)); // Alış fiyatı
    komut.Parameters.AddWithValue("@satisfiyati", double.Parse(SatisFiyatitxt.Text)); // Satış fiyatı
    komut.Parameters.AddWithValue("@urunadi", ÜrünAditxt.Text); // Ürün adı

    // SQL komutunu veritabanında çalıştırıyoruz
    komut.ExecuteNonQuery();

    baglanti.Close(); // Veritabanı bağlantısını kapat

    daset.Tables["urun"].Clear(); // DataSet içindeki "urun" tablosunu temizliyoruz
    ÜrünListele(); // Ürünleri tekrar listeliyoruz

    MessageBox.Show("Güncelleme yapıldı"); // Kullanıcıya bilgilendirme mesajı gösteriyoruz

    // Formdaki tüm TextBox bileşenlerini temizliyoruz
    foreach (Control item in this.Controls)
    {
        if (item is TextBox)
        {
            item.Text = "";
        }
    }
}
```

```

private void ÜrünListele()
{
    baglanti.Open(); // Veritabanı bağlantısını aç

    // SqlDataAdapter ile ürün tablosundaki tüm verileri çekiyoruz
    SqlDataAdapter adtr = new SqlDataAdapter("select * from urun", baglanti);
    adtr.Fill(daset, "urun"); // DataTable içine verileri dolduruyoruz
    dataGridView2.DataSource = daset.Tables["urun"]; // DataGridView'e verileri bağliyoruz

    baglanti.Close(); // Veritabanı bağlantısını kapat
}

1 bagvuru
private void dataGridView2_CellDoubleClick(object sender, DataGridViewCellEventArgs e)
{
    // DataGridView'de bir hücreye çift tıklandığında yapılacak işlemler
    // Seçilen ürünün bilgilerini ilgili TextBox'lara dolduruyoruz
    BarkodNottxt.Text = dataGridView2.CurrentRow.Cells["barkodno"].Value.ToString();
    kategoritxt.Text = dataGridView2.CurrentRow.Cells["kategori"].Value.ToString();
    Markatxt.Text = dataGridView2.CurrentRow.Cells["marka"].Value.ToString();
    AlisFiyatitxt.Text = dataGridView2.CurrentRow.Cells["alisfiyati"].Value.ToString();
    SatisFiyatitxt.Text = dataGridView2.CurrentRow.Cells["satisfiyati"].Value.ToString();
    ÜrünAditxt.Text = dataGridView2.CurrentRow.Cells["urunadi"].Value.ToString();
}

1 bagvuru
private void button1_Click(object sender, EventArgs e)
{
    if (BarkodNottxt.Text != "")
    {
        baglanti.Open(); // Veritabanı bağlantısını aç

        // Seçilen ürünün kategori ve markasını güncellemek için SQL UPDATE komutu kullanıyoruz
        SqlCommand komut = new SqlCommand("update urun set kategori=@kategori,marka=@marka where barkodno=@barkodno ", baglanti);
        komut.Parameters.AddWithValue("@barkodno", BarkodNottxt.Text); // Barkod numarası
        komut.Parameters.AddWithValue("@kategori", kombokategori.Text); // Yeni kategori
        komut.Parameters.AddWithValue("@marka", combomarka.Text); // Yeni marka

        komut.ExecuteNonQuery(); // SQL komutunu çalıştır

        baglanti.Close(); // Veritabanı bağlantısını kapat

        MessageBox.Show("Güncelleme yapıldı"); // Kullanıcıya güncelleme yapıldı mesajı göster

        daset.Tables["urun"].Clear(); // DataSet içindeki "urun" tablosunu temizle
        ÜrünListele(); // Güncellenmiş ürün listesini tekrar yükle
    }
    else
    {
        MessageBox.Show("Barkod no yazılı değil!"); // Hata mesajı
    }
}

private void kombokategori_SelectedIndexChanged(object sender, EventArgs e)
{
    // ComboBox'ın seçilen kategorisine göre marka listesini güncelleyen metod
    combomarka.Items.Clear(); // ComboBox içeriğini temizle
    combomarka.Text = ""; // ComboBox'ın seçili değerini temizle

    baglanti.Open(); // Veritabanı bağlantısını aç

    // Seçilen kategoriye göre marka bilgilerini getiren SQL sorgusu
    SqlCommand komut = new SqlCommand("select kategori,marka from markabilgileri where kategori='" + kombokategori.SelectedItem + "'", baglanti);
    SqlDataReader read = komut.ExecuteReader(); // SQL sorgusunu çalıştır

    // Okunan verileri ComboBox'a ekleyerek marka listesini güncelle
    while (read.Read())
    {
        combomarka.Items.Add(read["marka"].ToString());
    }

    baglanti.Close(); // Veritabanı bağlantısını kapat
}

1 bagvuru
private void sil_Click(object sender, EventArgs e)
{
    baglanti.Open(); // Veritabanı bağlantısını aç

    // DataGridView'de seçilen ürünü silen SQL sorgusu
    SqlCommand komut = new SqlCommand("delete from urun where barkodno='" + dataGridView2.CurrentRow.Cells["barkodno"].Value.ToString() + "'", baglanti);
    komut.ExecuteNonQuery(); // SQL sorgusunu çalıştır

    baglanti.Close(); // Veritabanı bağlantısını kapat

    daset.Tables["urun"].Clear(); // DataSet içindeki "urun" tablosunu temizle
    ÜrünListele(); // Ürünleri tekrar listele

    MessageBox.Show("Kayıt Silindi"); // Kullanıcıya bilgilendirme mesajı göster
}

1 bagvuru
private void textBox1_TextChanged(object sender, EventArgs e)
{
    DataTable tablo = new DataTable(); // Yeni bir DataTable oluştur

    baglanti.Open(); // Veritabanı bağlantısını aç

    // Barkod numarasına göre ürünleri getiren SQL sorgusu
    SqlDataAdapter adtr = new SqlDataAdapter("select * from urun where barkodno like '%" + txtBarkodNogoreAra.Text + "%'", baglanti);
    adtr.Fill(tablo); // DataTable'ı doldur
    dataGridView2.DataSource = tablo; // DataGridView'e DataTable'ı bağla

    baglanti.Close(); // Veritabanı bağlantısını kapat
}

```

Burada eklediğimiz ürünlerin listesi bulunmakta. Barkod kodu girildiğinde ya da listede ürün seçildiğinde ürünün bütün bilgileri otomatik doldurulur. Sadece marka ve kategorinin güncellemesini aşağıda farklı koydum. Onları combobox ile yaptığım için güncelleme kategoriside farklı . barkod kodundan arandığında listede aranılan barkod gözükmekte. Seçilen ürün yine sil butonunda sbasıldığında listeden silinmektedir.

MÜŞTERİ EKLE

```
public partial class frmMüşteriEkle : Form
{
    1 bayvuru
    public frmMüşteriEkle()
    {
        InitializeComponent();
    }

    // SqlConnection nesnesi oluşturarak veritabanı bağlantısını tanımlıyoruz
    SqlConnection baglanti = new SqlConnection("Data Source=FEDERAL\\SQLEXPRESS01;Initial Catalog=musteriListes;Integrated Security=True");

    1 bayvuru
    private void frmMüşteriEkle_Load(object sender, EventArgs e)
    {
        // Form yüklendiğinde yapılacak işlemler buraya yazılır
    }

    1 bayvuru
    private void button1_Click(object sender, EventArgs e)
    {
        // "Ekle" butonuna tıklandığında yapılacak işlemler burada gerçekleştirilir

        baglanti.Open(); // Veritabanı bağlantısını aç

        // SQL INSERT komutu ile müşteri bilgilerini veritabanına ekliyoruz
        SqlCommand komut = new SqlCommand("insert into müşteri(tc,adsoyad,telefon,adres,email) values(@tc,@adsoyad,@telefon,@adres,@email)", baglanti);

        // Parametrelerimize değerleri atıyoruz
        komut.Parameters.AddWithValue("@tc", txtTc.Text); // TC Kimlik No
        komut.Parameters.AddWithValue("@adsoyad", txtAdSoyad.Text); // Ad ve soyad
        komut.Parameters.AddWithValue("@telefon", txtTelefon.Text); // Telefon numarası
        komut.Parameters.AddWithValue("@adres", txtAdres.Text); // Adres
        komut.Parameters.AddWithValue("@email", txtEmail.Text); // E-posta

        komut.ExecuteNonQuery(); // SQL komutunu veritabanında çalıştır

        baglanti.Close(); // Veritabanı bağlantısını kapat

        MessageBox.Show("Müşteri kaydı eklendi"); // Kullanıcıya bilgilendirme mesajı göster

        // Formdaki tüm TextBox bileşenlerini temizliyoruz
        foreach (Control item in this.Controls)
        {
            if (item is TextBox)
            {
                item.Text = "";
            }
        }
    }
}
```

Burada satıcının müşterilerinden eklemesini sağlar. Eklenen bilgiler müşteri listesine gider.

MÜŞTERİ LİSTESİ

TC Ara

Sa

TC
Ad Soyad
Telefon
Adres
Email

Güncelle

```

public partial class frmMusteriListele : Form
{
    1 bagvuru
    public frmMusteriListele()
    {
        InitializeComponent(); // Form yüklendiğinde otomatik olarak çağrılan metod. Form bileşenlerini başlatır.
    }

    SqlConnection baglanti = new SqlConnection("Data Source=FEDERAL\\SQLEXPRESS01;Initial Catalog=musteriListes;Integrated Security=True");
    DataSet daset = new DataSet();

    1 bagvuru
    private void frmMusteriListele_Load(object sender, EventArgs e)
    {
        Kayit_Goster(); // Form yüklendiğinde Kayit_Goster metodu çağrılır ve veriler DataGridView'e yüklenir.
    }

    2 bagvuru
    private void Kayit_Goster()
    {
        baglanti.Open();
        // SqlDataAdapter nesnesi tanımlanıyor. Veritabanından veri çekmek için kullanılır.
        SqlDataAdapter adr = new SqlDataAdapter("select *from müşteri", baglanti);
        adr.Fill(daset, "müşteri");// SqlDataAdapter ile veritabanından veriler DataSet'e yükleniyor. "müşteri" tablosu DataSet içinde "müşteri" adında bir tabloya aktarılıyor.
        // DataGridView kontrolüne DataSet içindeki "müşteri" tablosu bağlanarak veriler gösteriliyor.
        dataGridView1.DataSource = daset.Tables["müşteri"];
        baglanti.Close(); // Veritabanı bağlantısı kapatılıyor.
    }

    1 bagvuru
    private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
    {
        // DataGridView hücre içeriği tıklandığında herhangi bir işlem yapılmasına gerek olmadığı için bu metod boş bırakılmış.
    }

    1 bagvuru
    private void dataGridView1_CellDoubleClick(object sender, DataGridViewCellEventArgs e)
    {
        // DataGridView hücreleri çift tıklandığında bu metod çalışır.

        // Seçilen müşterinin bilgileri ilgili TextBox kontrollerine aktarılır.
        txtTc.Text = dataGridView1.CurrentRow.Cells["tc"].Value.ToString();
        txtAdSoyad.Text = dataGridView1.CurrentRow.Cells["adsoyad"].Value.ToString();
        txtTelefon.Text = dataGridView1.CurrentRow.Cells["telefon"].Value.ToString();
        TxtAdres.Text = dataGridView1.CurrentRow.Cells["adres"].Value.ToString();
        TxtEmail.Text = dataGridView1.CurrentRow.Cells["email"].Value.ToString();
    }

    1 bagvuru
    private void btnGuncelle_Click(object sender, EventArgs e)
    {
        baglanti.Open();
        SqlCommand komut = new SqlCommand("update müşteri set adsoyad=@adsoyad,telefon=@telefon,adres=@adres,email=@email where tc=@tc", baglanti);

        // Parametreler atanır.
        komut.Parameters.AddWithValue("@tc", txtTc.Text);
        komut.Parameters.AddWithValue("@adsoyad", txtAdSoyad.Text);
        komut.Parameters.AddWithValue("@telefon", txtTelefon.Text);
        komut.Parameters.AddWithValue("@adres", TxtAdres.Text);
        komut.Parameters.AddWithValue("@email", TxtEmail.Text);

        // Komut veritabanında çalıştırılır.
        komut.ExecuteNonQuery();
        baglanti.Close();

        // DataSet içindeki "müşteri" tablosu temizlenir.
        daset.Tables["müşteri"].Clear();
        // DataGridView'e güncel veriler yeniden yüklenir.
        Kayit_Goster();
        // Kullanıcıya güncelleme yapıldı mesajı gösterilir.
        MessageBox.Show("Müşteri kaydı güncellendi");
        // Formdaki TextBox kontrolleri temizlenir.
        foreach (Control item in this.Controls)
        {
            if (item is TextBox)
            {
                item.Text = "";
            }
        }
    }
}

```

```

}
private void btnSil_Click(object sender, EventArgs e)
{
    baglanti.Open();
    // Silme işlemi için SqlCommand nesnesi oluşturuluyor. "delete from müşteri" sorgusu kullanılıyor ve silinecek TC değeri parametre olarak ekleniyor.
    SqlCommand komut = new SqlCommand("delete from müşteri where tc=" + dataGridView1.CurrentRow.Cells["tc"].Value.ToString() + "", baglanti);
    // SqlCommand ile veritabanında silme işlemi gerçekleştiriliyor.
    komut.ExecuteNonQuery();
    baglanti.Close();
    MessageBox.Show("Kayıt Silindi");// Kullanıcıya kaydın silindiğine dair bilgi veren bir mesaj gösteriliyor.
}

1 bayvuru
private void txtTcAra_TextChanged(object sender, EventArgs e)
{
    // DataTable nesnesi oluşturuluyor.
    DataTable tablo = new DataTable();
    baglanti.Open();
    // SqlDataAdapter nesnesi oluşturuluyor ve müşteri tablosundan TC alanı içinde arama yapılıyor.
    SqlDataAdapter adtr = new SqlDataAdapter("select *from müşteri where tc like '%" + txtTcAra.Text + "%'", baglanti);

    // Veriler SqlDataAdapter ile DataTable'a aktarılıyor.
    adtr.Fill(tablo);

    // DataGridView'e DataTable içindeki veriler bağlanıyor.
    dataGridView1.DataSource = tablo;
    baglanti.Close();
}
}

```

Müşteri ekle sayfasında eklenen müşterilerin listesi ekranda gözüktür.
 Yine listedeki seçtiğiniz satır otomatik doldurulur girdiğiniz tc ile de otomatik doldurulur.
 Değiştirmek istenen bilgiler güncelle butonuyla değişir.
 Tc ara kısmında girdiğiniz tceye ait satırlar sadece listede gözüktür.
 Ve yine silme butonuyla seçilen satır silinir.

KATEGORİ EKLEME SAYFASI

```
public partial class frmkategori : Form
{
    1 bagvuru
    3 public frmkategori()
    {
        InitializeComponent();
    }
    // Veritabanı bağlantısı için SqlConnection nesnesi oluşturulur.
    SqlConnection baglanti = new SqlConnection("Data Source=FEDERAL\\SQLEXPRESS01;Initial Catalog=musteriListes;Integrated Security=True");

    // "Ekle" butonuna tıklandığında çalışacak metod.
    1 bagvuru
    3 private void button1_Click(object sender, EventArgs e)
    {
        // Veritabanı bağlantısı açılır.
        baglanti.Open();

        // SQL sorgusu ile kategori tablosuna yeni bir kategori eklenir.
        SqlCommand komut = new SqlCommand("insert into kategori(kategori) values ('" + textBox1.Text + "')", baglanti);

        // Komut veritabanında çalıştırılır.
        komut.ExecuteNonQuery();

        // Veritabanı bağlantısı kapatılır.
        baglanti.Close();

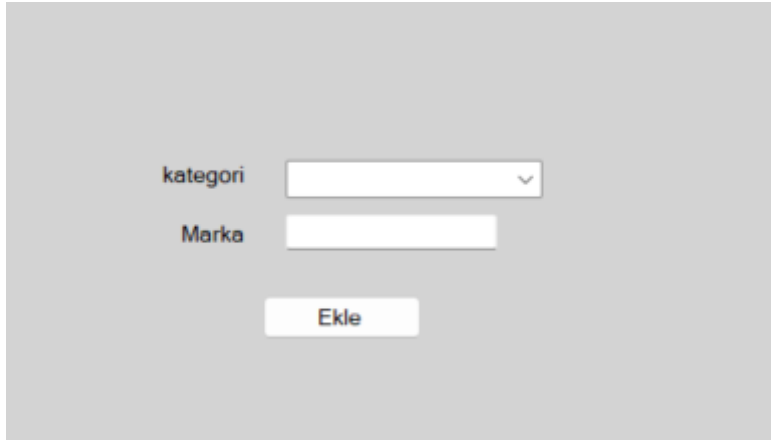
        // TextBox içeriği temizlenir.
        textBox1.Text = "";

        // Kullanıcıya bilgi mesajı gösterilir.
        MessageBox.Show("Kategoriye eklendi");
    }

    // Form yüklendiğinde çalışacak metod.
    1 bagvuru
    3 private void frmkategori_Load(object sender, EventArgs e)
    {
        // Herhangi bir işlem yapılmadığı için bu metod boş bırakılmış.
    }
}
```

BURADA SİSTEME YENİ KATEGORİ BİLGİLERİ EKLENİR.

MARKA EKLEME SAYFASI

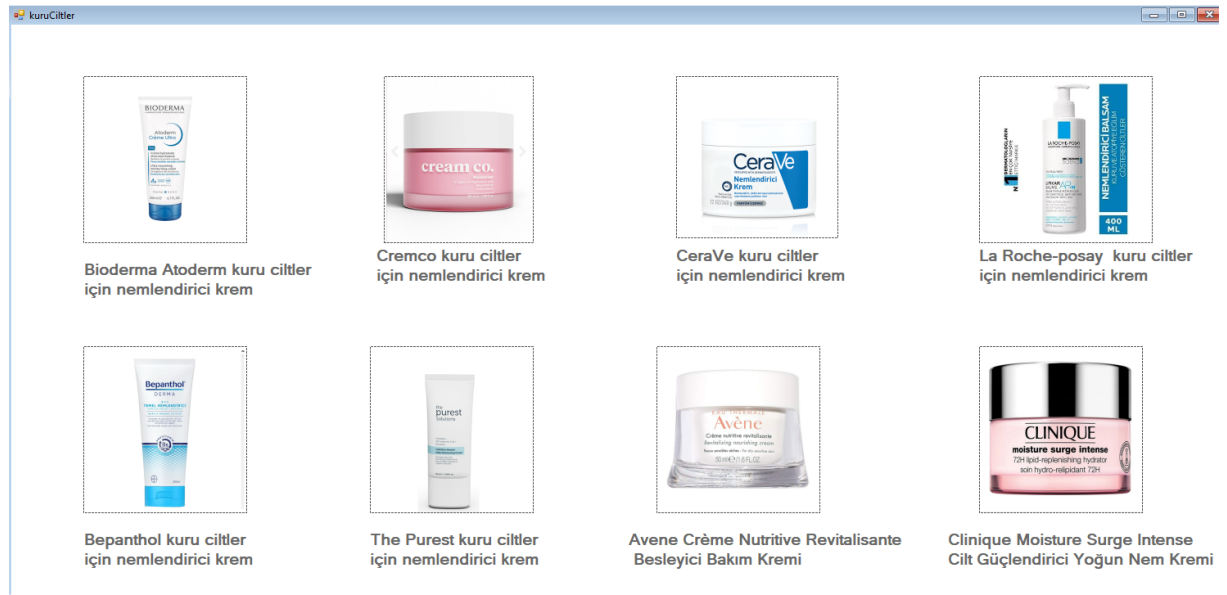


```
public partial class frmMarka : Form
{
    1baguru
    public frmMarka()
    {
        InitializeComponent();
    }
    SqlConnection baglanti = new SqlConnection("Data Source=FEDERAL\\SQLEXPRESS01;Initial Catalog=musteriListes;Integrated Security=True");
    1baguru
    private void button1_Click(object sender, EventArgs e)
    {
        baglanti.Open();
        // SQL sorgusu ile markabilgileri tablosuna yeni bir marka eklenir.
        SqlCommand komut = new SqlCommand("insert into markabilgileri(kategori,marka) values ('" + comboBox1.Text + "', '" + textBox1.Text + "')", baglanti);
        // Komut veritabanında çalıştırılır.
        komut.ExecuteNonQuery();
        baglanti.Close();
        // TextBox ve ComboBox içerikleri temizlenir.
        textBox1.Text = "";
        comboBox1.Text = "";

        // Kullanıcıya bilgi mesajı gösterilir.
        MessageBox.Show("Marka eklendi");
    }
    // Kategorileri ComboBox'a getiren metod.
    1baguru
    private void kategoriGetir()
    {
        // Veritabanı bağlantısı açılır.
        baglanti.Open();
        // SQL sorgusu ile kategori tablosundan kategoriler alınır.
        SqlCommand komut = new SqlCommand("select *from kategori", baglanti);
        // Sorgu sonucu okuyucu oluşturulur.
        SqlDataReader read = komut.ExecuteReader();
        // Okuyucu üzerinde dönerek ComboBox'a kategoriler eklenir.
        while (read.Read())
        {
            comboBox1.Items.Add(read["kategori"].ToString());
        }
        // Veritabanı bağlantısı kapatılır.
        baglanti.Close();
    }
    // Form yüklendiğinde çalışacak metod.
    1baguru
    private void frmMarka_Load(object sender, EventArgs e)
    {
        // Kategorileri ComboBox'a getir.
        kategoriGetir();
    }
}
```

Girilen kategoriye göre marka oluşturup kaydedilir. kategoriGetir(); tanımladım ve tekrar yazmak yerine onu kullandım.

KULLANICI SAYFASI – KURU CİLTLER



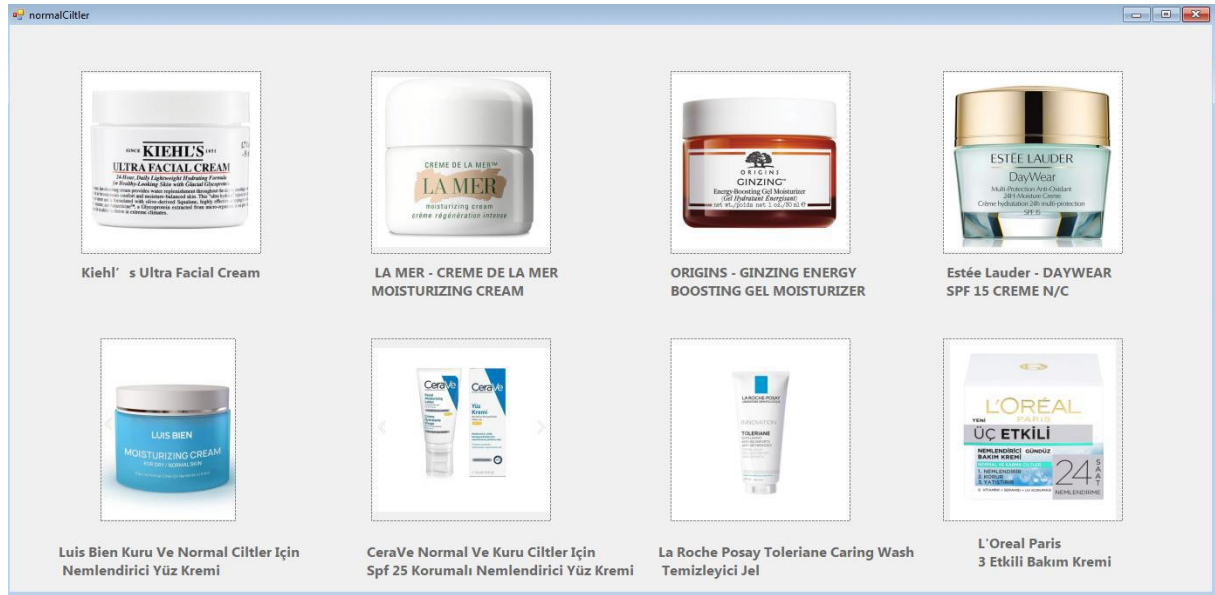
```
namespace SineBeauty
{
    4 başvuru
    public partial class kuruCiltler : Form
    {
        1 başvuru
        public kuruCiltler()
        {
            InitializeComponent();

            // PictureBox'a tıklama olayı için tanımlanan metot.
            1 başvuru
            private void pictureBox1_Click(object sender, EventArgs e)
            {
                // Kod burada PictureBox'a tıklandığında gerçekleşecek işlemleri içermelidir.
                // Ancak şu an için metot boş bırakılmış.
            }

            // Label'a tıklama olayı için tanımlanan metot.
            1 başvuru
            private void label7_Click(object sender, EventArgs e)
            {
                // Kod burada Label'a tıklandığında gerçekleşecek işlemleri içermelidir.
                // Ancak şu an için metot boş bırakılmış.
            }

            // Form yüklendiğinde gerçekleşecek işlemler için tanımlanan metot.
            1 başvuru
            private void kuruCiltler_Load(object sender, EventArgs e)
            {
                // Kod burada form yüklendiğinde gerçekleşecek işlemleri içermelidir.
                // Ancak şu an için metot boş bırakılmış.
            }
        }
    }
}
```

Kuru ciltler için kremler sayfası.



```
namespace SNbeauty
{
    4 başvuru
    public partial class normalCiltler : Form
    {
        1 başvuru
        public normalCiltler()
        {
            InitializeComponent();
        }

        // Label'a tıklama olayı için tanımlanan metot.
        1 başvuru
        private void label7_Click(object sender, EventArgs e)
        {
            // Kod burada Label'a tıklandığında gerçekleşecek işlemleri içermelidir.
            // Ancak şu an için metot boş bırakılmış.
        }

        // Form yüklendiğinde gerçekleşecek işlemler için tanımlanan metot.
        1 başvuru
        private void normalCiltler_Load(object sender, EventArgs e)
        {

```

Normal ciltler için uygun olan kremler sayfası.