



GÖRSEL PROGRAMLAMA PROJE  
RAPORU

YAZILIM MÜHENDİSLİĞİ  
FET309 GÖRSEL PROGRAMLAMA

ŞULE KARTAL  
22040301049

## İÇİNDEKİLER:

- 1-Projenin konusu ve amacı
- 2-Projede kullanılan yapılar ve kodlar
- 3-Veritabanı Yapısı

**Proje Özeti :** Ben projemde teknoloji mağazası yaptım. Personel ve kullanıcı girişi olarak iki ayrı kısımdan oluşuyor. Personel kısmına girince ürünleri listele, ekle, sil, güncelle kısmı var. Kullanıcı kısmında ise o ürünlerden istediğini seçip sepete ekleyip sipariş verebiliyor. Veritabanı olarak MongoDB kullandım.

**Projenin Konusu ve Amacı :** Projenin amacına gelecek olursak proje ilk başta personel, kullanıcı girişi için iki bölüme ayrılıyor. Personel kısmına girince yeni ürün ekleyebilir, ürün silebilir, güncelleyebilir ve listeleyip bütün ürünleri görüntüleyebilir. Kullanıcı girişinde ise veritabanında eklenen ürünleri görüntüleyip almak istediği ürünü seçip sepete ekleyip siparişini onaylayabilir. Sipariş onayla kısmında ad-soyad, adres ve kart bilgileri gibi bilgileri doldurup siparişini onaylayabilir. Müşteri eğer üye değilse giriş ekranında üye ol kısmından üyelik oluşturabilir. Ve bu işlemlerin hepsi veritabanına kaydedilir.

**Projede Kullanılan Yapılar ve Kodlar :**

## Giriş sayfası :

- Burda üç tane buton bulunur Personel girişi, Kullanıcı girişi ve kaydol



## Personel Giriş Sayfası:

- Burda veritabanına kaydedilen personellerin kullanıcı adı ve parola bilgileri ile giriş yapılır.



```
using MongoDB.Bson;
using MongoDB.Driver;
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 Teknoloji_Takip_Sistemi
{
    public partial class personel_giris : Form
    {
        string connectionString = "mongodb://localhost:27017";
        string databaseName = "teknoloji_takip_sistemi";
        string collectionName = "personel";

        private IMongoDatabase database;
        private IMongoCollection<BsonDocument> collection;
        public personel_giris()
        {
            InitializeComponent();
            var client = new MongoClient("mongodb://localhost:27017");
            database = client.GetDatabase("teknoloji_takip_sistemi");
            collection = database.GetCollection<BsonDocument>("personel");
        }

        private void button1_Click(object sender, EventArgs e)
        {
            // kullanıcı adı ve parola bilgisini textboxlara yazdırıyoruz
            string kullanıcı_Adi = textBox_kullanici_adi.Text.Trim();
            string parola = textBox_parola.Text.Trim();

            if (string.IsNullOrEmpty(kullanici_Adi) ||
                string.IsNullOrEmpty(parola))
            {
                MessageBox.Show("Lütfen kullanıcı adı ve şifre giriniz!",
                    "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }

            // MongoDB'de kullanıcıyı kontrol etme
            var filter = Builders<BsonDocument>.Filter.Eq("kullanici_adi",
                kullanıcı_Adi) & Builders<BsonDocument>.Filter.Eq("parola", parola);
            var user = collection.Find(filter).FirstOrDefault();
            // burda girilen bilgiler veritabanındaki ile aynıysa giriş
            yapar
            if (user != null)
            {
                MessageBox.Show("Giriş başarılı!", "Bilgi",
                    MessageBoxButtons.OK, MessageBoxIcon.Information);
                personel gecis = new personel();
                gecis.Show();
                this.Hide();
            }
            // eğer aynı değilse hata mesajı verir
            else
            {
                MessageBox.Show("Kullanıcı adı veya şifre yanlış!", "Hata",
                    MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}

```

```
}  
  
private void button_geri_Click(object sender, EventArgs e)  
{  
    Form1 gecis = new Form1();  
    gecis.Show();  
    this.Hide();  
}  
}
```

## Personel Sayfası:

- Burda ürün bilgilerini ekle, sil, güncelle, listele yapabilir

The screenshot shows a Windows application window titled "personel". The window has a dark blue background with a glowing Apple logo in the center. On the left side, there are eight text input fields with labels: "İd", "Marka", "Ürün Adı", "Kategori", "Fiyat", "Depolama Alanı", "Garanti", and "Renk". On the right side, there are five buttons: "Ürün Ekle", "Ürün Sil", "Ürün Güncelle", "Listele", and "Geri". The bottom half of the window is a large, empty gray area.

personel

id: 5

Marka: apple

Ürün Adı: iphone 16 pro

Kategori: telefon

Fiyat: 82999

Depolama Alanı: 256gb

Garanti: 2 yıl

Renk: çöl titanyum

Ürün Ekle

Ürün Sil

Ürün Güncelle

Listele

Geri

	id	marka	urun_adi	kategori	fiyat	depolama_a
	1	apple	iphone 16 pro max	telefon	99999	512gb
	2	apple	iphone 13	telefon	33999	256gb
	3	apple	iphone 7 plus	telefon	5000	256gb
	4	apple	iphone 12	telefon	28999	128gb
▶	5	apple	iphone 16 pro	telefon	82999	256gb
	6	apple	iphone 16	telefon	64999	128gb
	7	apple	watch series 10	akıllı saat	18299	-

Listeleye tıklayınca alt kısımda gözüküyor ürünün üzerine tıklayınca üst taraftaki gibi textboxlar içine yazılır ve oradan yapmak istediğimiz işlemi seçeriz.

```
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 MongoDB.Bson;
using MongoDB.Driver;

namespace Teknoloji_Takip_Sistemi
{
    public partial class personel : Form
    {
        string connectionString = "mongodb://localhost:27017"; // yerel mongodb için
        string databaseName = "teknoloji_takip_sistemi";
        string collectionName = "urunler";

        private IMongoDatabase database;
        private IMongoCollection<BsonDocument> collection;
```

```

public personel()
{
    InitializeComponent();
    var client = new MongoClient("mongodb://localhost:27017");
    database = client.GetDatabase("teknoloji_takip_sistemi");
    collection = database.GetCollection<BsonDocument>("urunler");
    displayUrunler();
}

private void personel_Load(object sender, EventArgs e)
{
    dataGridView1.DataSource = null;
}

private void button_urun_ekle_Click(object sender, EventArgs e)
{
    // burda yeni ürün eklemek için için textboxlara yazdırıyoruz
    try
    {
        var yeni_urun = new BsonDocument
        {
            {"urun_id", textBox_id.Text.Trim()},
            {"marka", textBox_marka.Text.Trim()},
            {"urun_adi", textBox_urun_adi.Text.Trim()},
            {"kategori", textBox_kategori.Text.Trim()},
            {"fiyat", textBox_fiyat.Text.Trim()},
            {"depolama_alani", textBox_depolama_alani.Text.Trim()},
            {"garanti", textBox_garanti.Text.Trim()},
            {"renk", textBox_renk.Text.Trim()}
        };
        // insertone ile ürünü ekleriz
        collection.InsertOne(yeni_urun);
        MessageBox.Show("Ürün başarıyla eklendi.");

        ClearUrun();
        displayUrunler();
    }
    catch (Exception ex) {
        MessageBox.Show("Hata" + ex.Message, "Hata Mesajı",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

public void displayUrunler()
{
    // burda ürünleri gerekli alana yazdırırız
    try
    {
        var urunler = collection.Find(new BsonDocument()).ToList();
        var urun_listesi = urunler.Select(k => new
        {
            id = k.Contains("urun_id") ? k["urun_id"].ToString() : "",
            marka = k.Contains("marka") ? k["marka"].ToString() : "",
            urun_adi = k.Contains("urun_adi") ?
k["urun_adi"].ToString() : "",
            kategori = k.Contains("kategori") ?
k["kategori"].ToString() : "",
            fiyat = k.Contains("fiyat") ? k["fiyat"].ToString() : "",
            depolama_alani = k.Contains("depolama_alani") ?
k["depolama_alani"].ToString() : "",
            garanti = k.Contains("garanti") ? k["garanti"].ToString() :
"",
            renk = k.Contains("renk") ? k["renk"].ToString() : ""
        }).ToList();
    }
}

```



```

        dataGridView1.DataSource = urun_listesi;

    }
    catch (Exception ex)
    {
        MessageBox.Show("Hata: " + ex.Message, "Hata Mesajı",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }

}

private void button_listele_Click(object sender, EventArgs e)
{
    // burda da ürünleri datagridde gösteririz
    displayUrunler();
}

int i = 0;
private void dataGridView1_CellMouseClick(object sender,
DataGridViewCellMouseEventArgs e)
{
    // bu kısımda datagrid üzerinde olan ürünlere mouse ile tıklayınca
    textboxlara yazılmasını sağlarız
    i = e.RowIndex;

    textBox_id.Text = dataGridView1.Rows[i].Cells[0].Value.ToString();
    textBox_marka.Text =
dataGridView1.Rows[i].Cells[1].Value.ToString();
    textBox_urun_adi.Text =
dataGridView1.Rows[i].Cells[2].Value.ToString();
    textBox_kategori.Text =
dataGridView1.Rows[i].Cells[3].Value.ToString();
    textBox_fiyat.Text =
dataGridView1.Rows[i].Cells[4].Value.ToString();
    textBox_depolama_alani.Text =
dataGridView1.Rows[i].Cells[5].Value.ToString();
    textBox_garanti.Text =
dataGridView1.Rows[i].Cells[6].Value.ToString();
    textBox_renk.Text =
dataGridView1.Rows[i].Cells[7].Value.ToString();
}

private void ClearUrun()
{
    textBox_id.Text = "";
    textBox_marka.Text = "";
    textBox_urun_adi.Text = "";
    textBox_kategori.Text = "";
    textBox_fiyat.Text = "";
    textBox_depolama_alani.Text = "";
    textBox_garanti.Text = "";
    textBox_renk.Text = "";
}

private void button_urun_sil_Click(object sender, EventArgs e)
{
    // eğer bir ürün seçilmemişse öge seçin diye mesaj verir
    if(string.IsNullOrEmpty(textBox_id.Text) ||
string.IsNullOrEmpty(textBox_marka.Text) ||
string.IsNullOrEmpty(textBox_urun_adi.Text) ||
string.IsNullOrEmpty(textBox_kategori.Text) ||

```

```

        string.IsNullOrEmpty(textBox_fiyat.Text) ||
string.IsNullOrEmpty(textBox_depolama_alani.Text) ||
        string.IsNullOrEmpty(textBox_garanti.Text) ||
string.IsNullOrEmpty(textBox_renk.Text))
    {
        MessageBox.Show("Silinecek Öğe Seçin", "Hata Mesajı",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        DialogResult check = MessageBox.Show("Silmek istediğinize emin
misiniz ? :"+
        + textBox_urun_adi.Text + "?" , "Onay
Mesajı", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question);

        if (check == DialogResult.Yes) {
            try
            {
                var filter =
Builders<BsonDocument>.Filter.Eq("urun_adi", textBox_urun_adi.Text.Trim());
                // burda da deleteone ile ürünü sileriz
                var result = collection.DeleteOne(filter);

                if(result.DeletedCount > 0)
                {
                    MessageBox.Show("Ürün Başarıyla Silindi", "Bilgi
Mesajı", MessageBoxButtons.OK, MessageBoxIcon.Error);

                    displayUrunler();
                }
                else
                {
                    MessageBox.Show("Silinecek Kayıt Bulunamadı",
"Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
                }
                ClearUrun();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Hata:" + ex.Message, "Hata Mesajı",
MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
        else
        {
            MessageBox.Show("Silme işlemi iptal edildi.", "Bilgi
Mesajı", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

private void button_urun_guncelle_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(textBox_id.Text) || string.IsNullOrEmpty(textBox_marka.Te
xt) ||
        string.IsNullOrEmpty(textBox_urun_adi.Text) ||
string.IsNullOrEmpty(textBox_kategori.Text) ||
        string.IsNullOrEmpty(textBox_fiyat.Text) ||
string.IsNullOrEmpty(textBox_depolama_alani.Text) ||
        string.IsNullOrEmpty(textBox_garanti.Text) ||
string.IsNullOrEmpty(textBox_renk.Text))
    {

```

```

        MessageBox.Show("Lütfen tüm alanları doldurun", "Hata Mesajı",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
        return;
    }
    try
    {
        var filter = Builders<BsonDocument>.Filter.Eq("urun_adi",
        textBox_urun_adi.Text.Trim());

        var update = Builders<BsonDocument>.Update
            .Set("urun_id", textBox_id.Text.Trim())
            .Set("marka", textBox_marka.Text.Trim())
            .Set("urun_adi", textBox_urun_adi.Text.Trim())
            .Set("kategori", textBox_kategori.Text.Trim())
            .Set("fiyat", textBox_fiyat.Text.Trim())
            .Set("depolama_alani", textBox_depolama_alani.Text.Trim())
            .Set("garanti", textBox_garanti.Text.Trim())
            .Set("renk", textBox_renk.Text.Trim());

        // updateone komutu ile değiştirmek istediğimiz bilgiyi
        değiştiririz
        var result = collection.UpdateOne(filter, update);

        if (result.ModifiedCount > 0)
        {
            MessageBox.Show("Ürün başarıyla güncellendi.", "Bilgi
            Mesajı", MessageBoxButtons.OK, MessageBoxIcon.Error);
            displayUrunler();
            ClearUrun();
        }
        else
        {
            MessageBox.Show("Değişiklik yapılmadı.", "Bilgi Mesajı",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }

    }
    catch (Exception ex) {
        MessageBox.Show("Hata:" + ex.Message , "Hata Mesajı",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void button_geri_Click(object sender, EventArgs e)
{
    personel_giris gecis = new personel_giris();
    gecis.Show();
    this.Hide();
}
}

```

## Kullanıcı Giriş Sayfası :

- Burda da personel girişi gibi veritabanında olan bilgileri girerek giriş yapılır.



```
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 MongoDB.Bson;
using MongoDB.Driver;

namespace Teknoloji_Takip_Sistemi
{
    public partial class kullanici : Form
    {
        string connectionString = "mongodb://localhost:27017";
        string dbName = "teknoloji_takip_sistemi";
        string collectionName = "kullanici";

        private IMongoDatabase database;
        private IMongoCollection<BsonDocument> collection;
        public kullanici()
        {
            InitializeComponent();
            var client = new MongoClient("mongodb://localhost:27017");
            database = client.GetDatabase("teknoloji_takip_sistemi");
            collection = database.GetCollection<BsonDocument>("kullanici");
        }

        private void button_giris_yap_Click(object sender, EventArgs e)
        {
            //burda textboxlara kullanıcı adı ve parola bilgisi yazılır
            string kullanıcı_Adi = textBox_kullanici_adi.Text.Trim();
            string parola = textBox_parola.Text.Trim();

            if (string.IsNullOrEmpty(kullanıcı_Adi) ||
                string.IsNullOrEmpty(parola))
            {
                MessageBox.Show("Lütfen kullanıcı adı ve şifre giriniz!",
                    "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
                return;
            }

            // MongoDB'de kullanıcıyı kontrol eder ve aşağıdaki if ile doğruysa
            yazdırır
            var filter = Builders<BsonDocument>.Filter.Eq("kullanici_adi",
                kullanıcı_Adi) & Builders<BsonDocument>.Filter.Eq("parola", parola);
            var user = collection.Find(filter).FirstOrDefault();

            if (user != null)
```

```

        {
            MessageBox.Show("Giriş başarılı!", "Bilgi",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
            kullanıcı_ekrani gecis = new kullanıcı_ekrani();
            gecis.Show();
            this.Hide();
        }
        else
        {
            MessageBox.Show("Kullanıcı adı veya şifre yanlış!", "Hata",
                MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }

    private void button_geri_Click(object sender, EventArgs e)
    {
        Form1 gecis = new Form1();
        gecis.Show();
        this.Hide();
    }
}

```

## Kullanıcı Sayfası:

- Burda kullanıcı istediği ürünü seçip sepete ekleyebilir ve siparişi onayla butonuna tıklayıp onaylar.





Bu şekilde ürün seçip sepete tıklayınca üstteki gibi bir görüntü elde ediyoruz.

```
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 MongoDB.Bson;
using MongoDB.Driver;

namespace Teknoloji_Takip_Sistemi
{
    public partial class kullanici_ekrani : Form
    {
        string connectionString = "mongodb://localhost:27017"; // yerel mongodb
        string databaseName = "teknoloji_takip_sistemi";
        string collectionName = "urunler";

        private IMongoDatabase database;
        private IMongoCollection<BsonDocument> collection;
        public kullanici_ekrani()
        {
            InitializeComponent();
            var client = new MongoClient("mongodb://localhost:27017");
            database = client.GetDatabase("teknoloji_takip_sistemi");
            collection = database.GetCollection<BsonDocument>("urunler");
            displayUrunler();
        }
    }
}
```

```

public void displayUrunler()
{
    try
    {
        var urunler = collection.Find(new BsonDocument()).ToList();
        var urun_listesi = urunler.Select(k => new
        {
            id = k.Contains("urun_id") ? k["urun_id"].ToString() : "",
            marka = k.Contains("marka") ? k["marka"].ToString() : "",
            urun_adi = k.Contains("urun_adi") ?
k["urun_adi"].ToString() : "",
            kategori = k.Contains("kategori") ?
k["kategori"].ToString() : "",
            fiyat = k.Contains("fiyat") ? k["fiyat"].ToString() : "",
            depolama_alani = k.Contains("depolama_alani") ?
k["depolama_alani"].ToString() : "",
            garanti = k.Contains("garanti") ? k["garanti"].ToString() :
"",
            renk = k.Contains("renk") ? k["renk"].ToString() : ""
        }).ToList();

        dataGridView1.DataSource = urun_listesi;

    }
    catch (Exception ex)
    {
        MessageBox.Show("Hata: " + ex.Message, "Hata Mesajı",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void button_toplam_Click(object sender, EventArgs e)
{
    decimal toplamFiyat = 0;
    string selectedItems = ""; // Seçilen ürünlerin bilgilerini tutacak
    değişken

    foreach (DataGridViewRow row in dataGridView1.Rows)
    {
        if (Convert.ToBoolean(row.Cells[0].Value)) // Eğer checkbox
işaretlenmişse
        {

            string marka = row.Cells[2].Value.ToString(); // ürün
markasını al
            string urunAdi = row.Cells[3].Value.ToString(); // Ürün
adını al
            string kategori = row.Cells[4].Value.ToString(); // ürün
kategorisini al
            decimal fiyat = Convert.ToDecimal(row.Cells[5].Value); //
Fiyatı al
            string depolama_alani = row.Cells[6].Value.ToString(); //
depolama alanı bilgisini al
            string garanti = row.Cells[7].Value.ToString(); // garanti
bilgisini al
            string renk = row.Cells[8].Value.ToString(); // renk
bilgisini al

            toplamFiyat += fiyat; // Fiyatı toplam fiyata ekle

```

```

        //Seçilen ürünlerin adlarını ve fiyatlarını alt alta ekle
        selectedItems += $"{marka} {urunAdi} {kategori}
{fiyat.ToString()}TL {depolama_alani} {garanti} {renk}\n";
    }
}

//Seçilen ürünler ve toplam fiyatı göster
if (string.IsNullOrEmpty(selectedItems))
{
    MessageBox.Show("Hiçbir ürün seçilmedi.");
}
else
{
    selectedItems += $"{\nToplam Fiyat: {toplamFiyat.ToString()}}";
    label_toplam.Text = selectedItems;
}

}

private void button_sepet_onay_Click(object sender, EventArgs e)
{
    siparis_onayil gecis = new siparis_onayil();
    gecis.Show();
    this.Hide();
}

private void button_geri_Click(object sender, EventArgs e)
{
    kullanıcı gecis = new kullanıcı();
    gecis.Show();
    this.Hide();
}

}
}

```

## Siparişi Onayla :

- Burda da istenilen bilgileri girerek siparişi onaylıyoruz.



siparis\_onayi1

Ad-Soyad	<input type="text"/>
Adres	<input type="text"/>
Telefon Numarası	<input type="text"/>
Kart üzerindeki Ad-Soyad	<input type="text"/>
Kart Numarası	<input type="text"/>
SKT	<input type="text"/>
CVV	<input type="text"/>

Siparişi Onayla

Geri

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

namespace Teknoloji_Takip_Sistemi
{
    public partial class siparis_onayil : Form
    {
        string connectionString = "mongodb://localhost:27017"; // yerel mongodb
        için
        string databaseName = "teknoloji_takip_sistemi";
        string collectionName = "siparis_bilgileri";

        private IMongoDatabase database;
        private IMongoCollection<BsonDocument> collection;
        public siparis_onayil()
        {
            InitializeComponent();
            var client = new MongoClient("mongodb://localhost:27017");
            database = client.GetDatabase("teknoloji_takip_sistemi");
            collection =
            database.GetCollection<BsonDocument>("siparis_bilgileri");
            displayBilgiler();
        }

        public void displayBilgiler()
        {
            // istenilen bilgileri textboxlara yazdırıyoruz
        }
    }
}
```

```

        string ad_soyad = textBox_ad_soyad.Text.Trim();
        string adres = textBox_adres.Text.Trim();
        string telefon = maskedTextBox_telefon.Text.Trim();
        string kart_ad = textBox_adsoyad.Text.Trim();
        string kart_numarasi = maskedTextBox_kart_numarasi.Text.Replace("
", "").Trim(); // Boşlukları kaldır
        string skt = maskedTextBox_skt.Text.Trim();
        string cvv = textBox_cvv.Text.Trim();

        // eğer herhangi bir alan boşsa hata mesajı verir
        if (string.IsNullOrEmpty(ad_soyad) ||
            string.IsNullOrEmpty(adres) ||
                string.IsNullOrEmpty(telefon) ||
            string.IsNullOrEmpty(kart_ad) ||
                string.IsNullOrEmpty(kart_numarasi) ||
            string.IsNullOrEmpty(skt) ||
                string.IsNullOrEmpty(cvv))
        {
            MessageBox.Show("Lütfen tüm alanları doldurun.", "Hata",
                MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }

        // kart numarasının boyut kontrolünü yapar
        if (kart_numarasi.Length != 16 || !long.TryParse(kart_numarasi, out
_))
        {
            MessageBox.Show("Kart numarası geçersiz.", "Hata",
                MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }

        if (cvv.Length != 3 || !int.TryParse(cvv, out _))
        {
            MessageBox.Show("CVV kodu geçersiz.", "Hata",
                MessageBoxButtons.OK, MessageBoxIcon.Warning);
            return;
        }

        try
        {
            var document = new BsonDocument
            {
                { "ad_soyad", ad_soyad },
                { "adres", adres },
                { "telefon_numarasi", telefon },
                { "kart_uzerindeki_ad_soyad", kart_ad },
                { "kart_numarasi", kart_numarasi },
                { "skt", skt },
                { "cvv", cvv }
            };

            collection.InsertOne(document); // Veriyi MongoDB'ye ekle
            MessageBox.Show("Sipariş başarıyla onaylandı.", "Başarılı",
                MessageBoxButtons.OK, MessageBoxIcon.Information);

            // Alanları temizle
            textBox_ad_soyad.Clear();
            textBox_adres.Clear();
            maskedTextBox_telefon.Clear();
            textBox_adsoyad.Clear();

```

```

        maskedTextBox_kart_numarasi.Clear();
        maskedTextBox_skt.Clear();
        textBox_cvv.Clear();
    }
    catch (Exception ex)
    {
        MessageBox.Show($"Veri kaydedilirken hata oluştu:
{ex.Message}", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
private void button_devam_Click(object sender, EventArgs e)
{
    displayBilgiler();
    // MessageBox.Show("Siparişiniz Onaylanmıştır.");
}

private void button_geri_Click(object sender, EventArgs e)
{
    kullanıcı_ekrani gecis = new kullanıcı_ekrani();
    gecis.Show();
    this.Hide();
}
}
}

```

## Yeni Kullanıcı Ekranı :

- Burda da istenilen bilgileri girip yeni bir kullanıcı oluşturuyoruz.



```

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 MongoDB.Bson;
using MongoDB.Driver;

namespace Teknoloji_Takip_Sistemi
{
    public partial class uye : Form
    {
        string connectionString = "mongodb://localhost:27017";
        string databaseName = "teknoloji_takip_sistemi";
        string collectionName = "kullanicilar";

        private IMongoDatabase database;
        private IMongoCollection<BsonDocument> collection;
        public uye()
        {
            InitializeComponent();
            var client = new MongoClient("mongodb://localhost:27017");
            database = client.GetDatabase("teknoloji_takip_sistemi");
            collection = database.GetCollection<BsonDocument>("kullanicilar");
        }

        private void button_uye_ol_Click(object sender, EventArgs e)
        {
            // alanların herhangi biri boşsa hata mesajı verir
            if (string.IsNullOrEmpty(textBox_ad.Text) ||
                string.IsNullOrEmpty(textBox_soyad.Text) ||
                string.IsNullOrEmpty(textBox_kullanici_adi.Text) ||
                string.IsNullOrEmpty(textBox_sifre.Text))
            {
                MessageBox.Show("Lütfen tüm alanları doldurun!");
            }

            // yeni kullanıcı oluşturmak için textboxlara yazdırılır
            var yeni_kullanici = new BsonDocument
            {
                {"ad", textBox_ad.Text.Trim() },
                {"soyad", textBox_soyad.Text.Trim() },
                {"kullanici_adi", textBox_kullanici_adi.Text.Trim() },
                {"parola", textBox_sifre.Text.Trim() }
            };

            collection.InsertOne(yeni_kullanici);
            MessageBox.Show("Yeni kullanıcı eklendi");

            textBox_ad.Clear();
            textBox_soyad.Clear();
            textBox_kullanici_adi.Clear();
            textBox_sifre.Clear();

            kullanıcı_gecis = new kullanıcı();
            gecis.Show();
            this.Hide();
        }

        private void button_geri_Click(object sender, EventArgs e)
        {

```

```
        Form1 gecis = new Form1();  
        gecis.Show();  
        this.Hide();  
    }  
}
```

## Veritabanı Yapısı :

### Kullanıcı Bilgileri

<pre>_id: ObjectId('676062c6b761073dd90d7baa') ad : "Seher" soyad : "Tombaş" kullanici_adi : "shrtmbs" parola : "seher123+"</pre>
<pre>_id: ObjectId('676062c6b761073dd90d7bab') ad : "Selma" soyad : "Çento" kullanici_adi : "selma_cento" parola : "selma2512"</pre>
<div><div><pre>_id: ObjectId('676062c6b761073dd90d7bac') ad : "Dilara" soyad : "Çento" kullanici_adi : "dil_ara" parola : "dilara9874"</pre></div><div>676062c6b761073dd90d7bac</div></div>

### Personel Bilgileri

```
_id: ObjectId('676058e4b846af91a9d3bb2f')
id: 1
kullanici_adi: "sule_1"
parola: "s1234"
```

```
_id: ObjectId('676058e4b846af91a9d3bb30')
id: 2
kullanici_adi: "esra_ctn"
parola: "esra1905"
```

```
_id: ObjectId('676058e4b846af91a9d3bb31')
id: 3
kullanici_adi: "_dilaraa_"
parola: "kurtul1515"
```

```
_id: ObjectId('676058e4b846af91a9d3bb32')
id: 4
kullanici_adi: "yagmurdvc."
parola: "yagmur00"
```

## Sipariş Bilgileri

```
_id: ObjectId('676344eedc09f51da911757d')
ad_soyad: "Şule Kartal"
adres: "ekrem çetin mahallesi doğan güreş paşa bulvarı no:64 kilis/merkez"
telefon_numarasi: "(552) 881-6268"
kart_uzerindeki_ad_soyad: "Şule Kartal"
kart_numarasi: "2598347512697586"
skt: "10.31"
cvv: "590"
```

## Ürün Bilgileri

```
_id: ObjectId('676198be97d03a081a0b6916')
urun_id : 1
marka : "apple"
urun_adi : "iphone 16 pro max"
kategori : "telefon"
fiyat : 99999
depolama_alani : "512gb"
garanti : "2 yıl"
renk : "siyah titanyum"
```

---

```
_id: ObjectId('676198be97d03a081a0b6917')
urun_id : "2"
marka : "apple"
urun_adi : "iphone 13"
kategori : "telefon"
fiyat : "33999"
depolama_alani : "256gb"
garanti : "2 yıl"
renk : "sarı"
```

---