

EGE ÜNİVERSİTESİ

Tire Kutsan Meslek Yüksekokulu

GÖRSEL PROGRAMLAMA

FİNAL ÖDEVİ



to do list app

Göksel Sina BİLECEN – 601800000018

içindekiler

Task Up nedir ?	3
Task Up nasıl kullanılır	3
Task Up ek özellikleri.....	4
Task Up uygulama görüntüleri.....	5
Task Up zorlandığım kısımlar.....	10
Task Up kullandığım metotlar.....	11
Task Up uygulama tasarımı.....	18
Kaynaklar.....	19

Task Up nedir ?

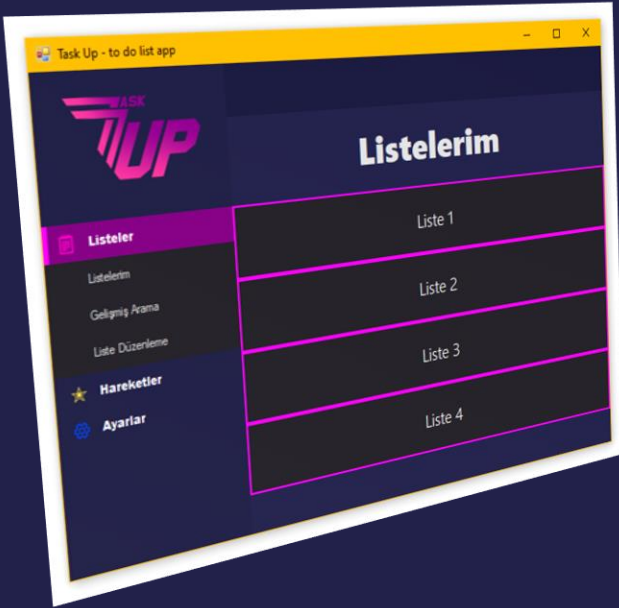
Task Up, yapacağınız işleri planlayarak bir liste haline getirebileceğiniz ve bu işlerinizi takip edebileceğiniz bir yardımcı araçtır.



Nasıl kullanılır ?

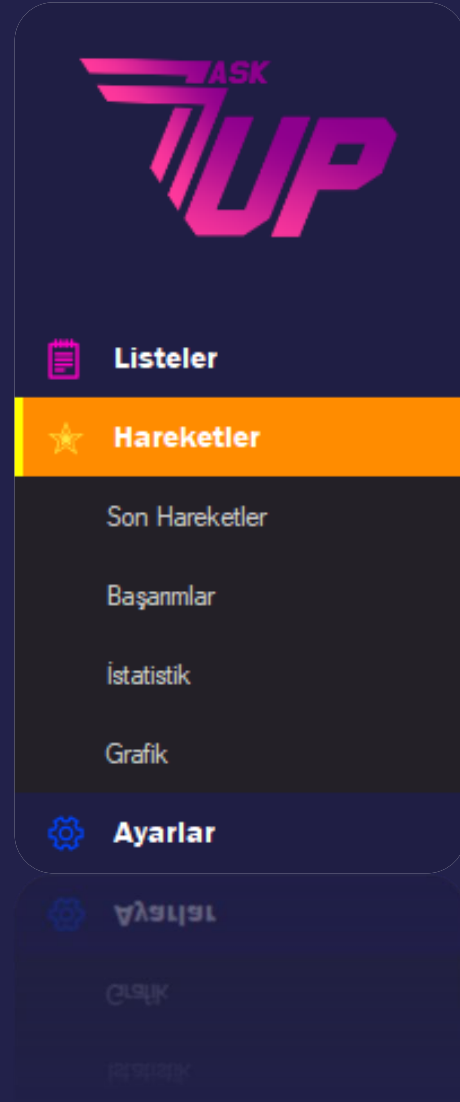
Kullanımı oldukça kolaydır.

Kullanıcıya 4 adet liste sağlanmıştır. Kullanıcı bu listelerin adını değiştirebilir, görev ekleyebilir, listedeki görevler üzerinde silme ve düzenleme işlemi yapabilir.

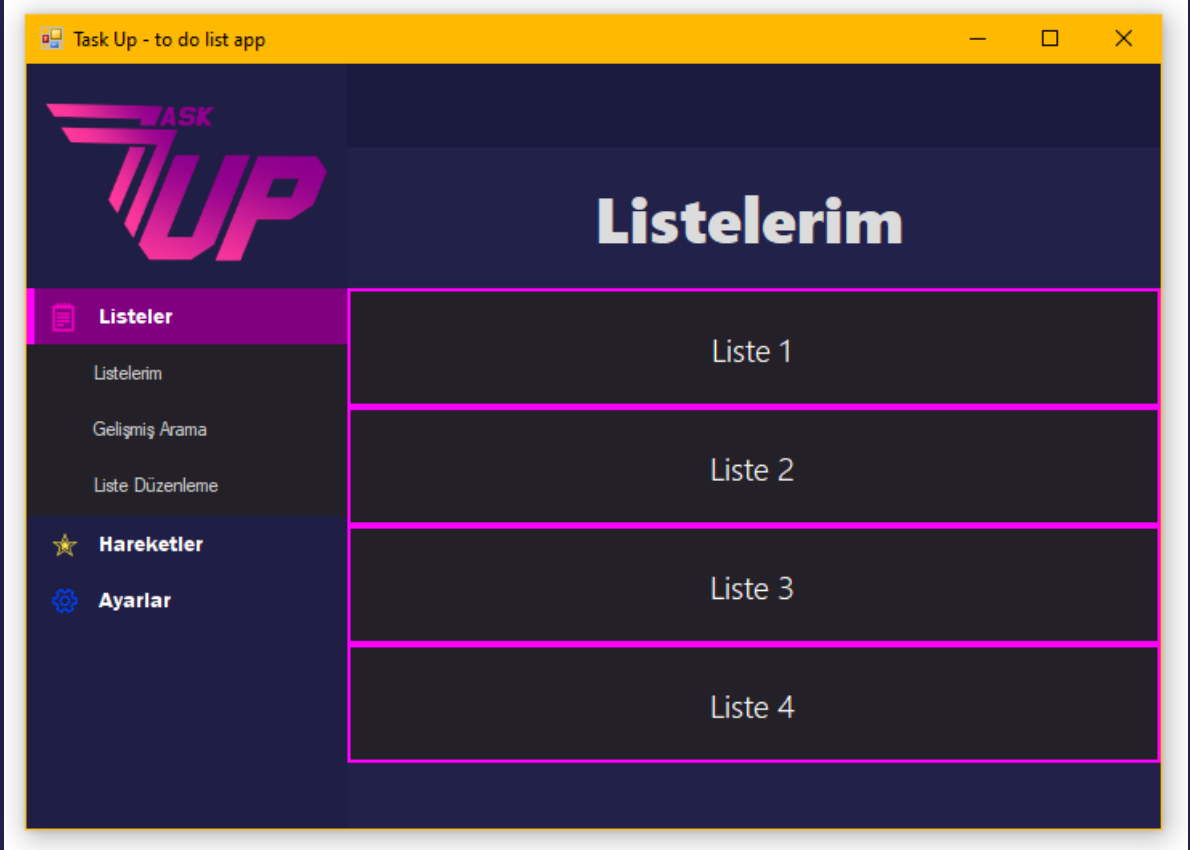


Başka ne gibi özellikleri bulunuyor ?

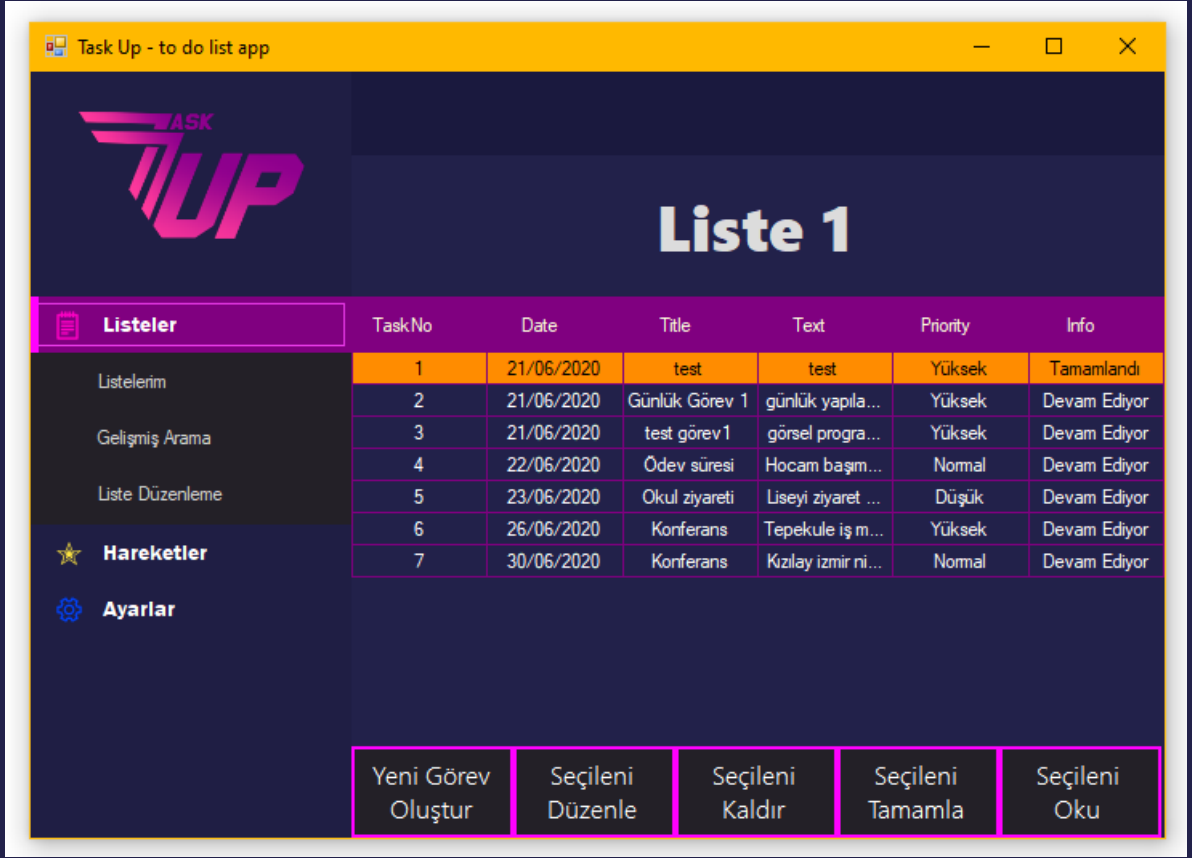
Task Up'ta son yapılan işlemleri kontrol edebilir, görev istatistiklerini görebilirsiniz. Tamamladığınız işler ile başarımlar kazanabilirsiniz. Hatta yaptığınız bu işleri grafik üzerinden takip edebilirsiniz. Verilerinizi yedekleyebilir, e posta adresinize yapılacaklar listenizi gönderebilirsiniz.



Uygulama Görüntüleri



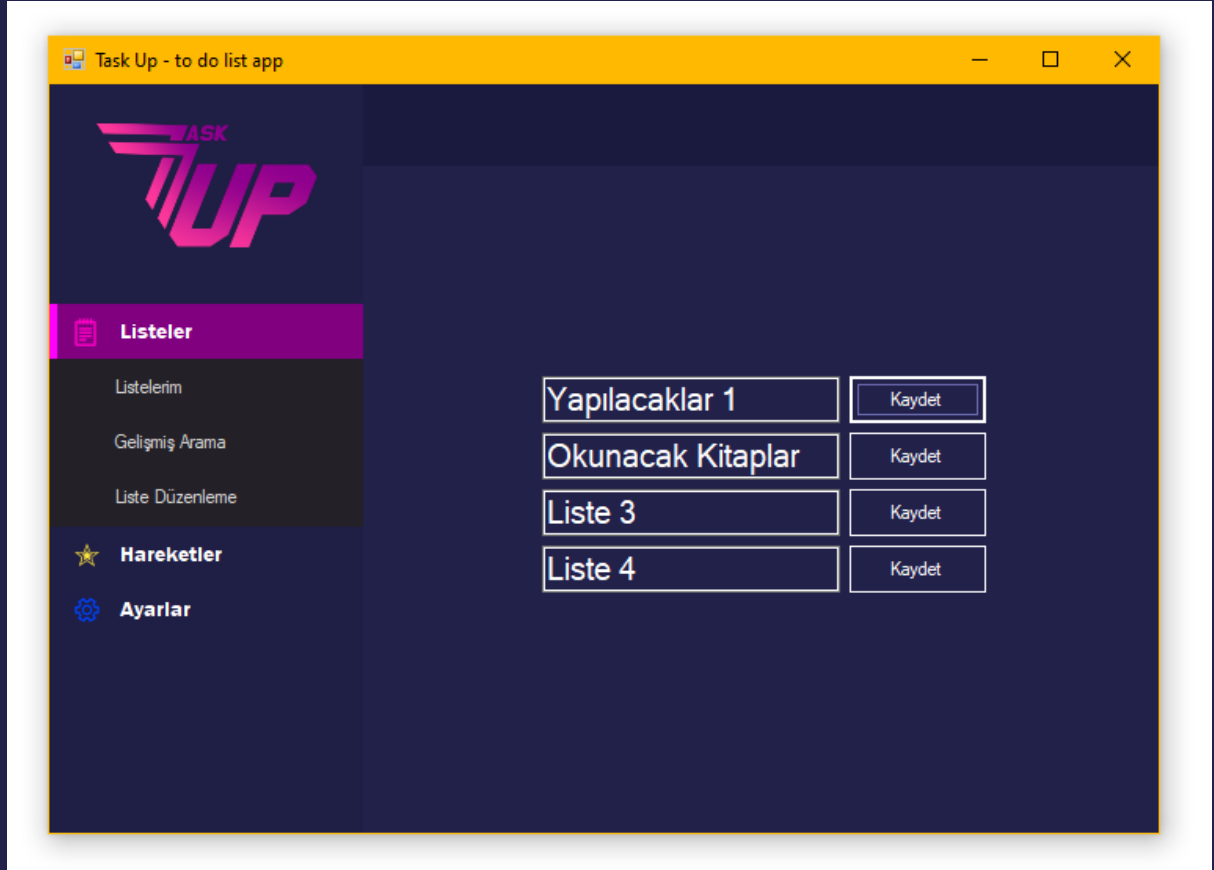
Listelerim : Uygulamanın giriş ekranı burasıdır. Kullanıcıdan verilen 4 listeden birini seçmesi beklenir. Yapacağı kayıtlar, seçtiği listeye eklenecektir. Bu liste adları değiştirilebilir.



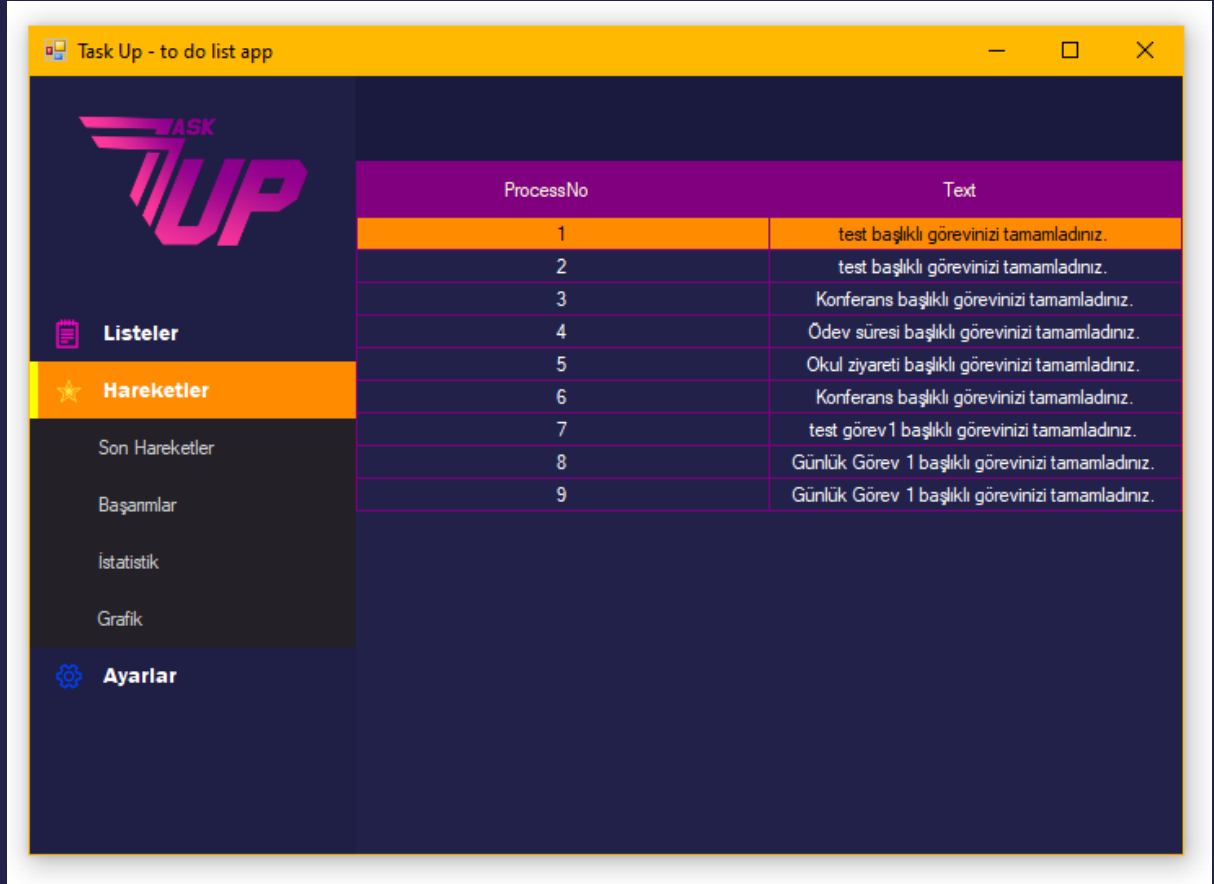
Liste Ekranı : Bu ekranda girilen kayıtlar listelenir. Alt bölümde 4 adet buton bulunur. Bu butonlar görev oluşturma, düzenleme, silme, tamamlama ve görev açıklaması okuma işlevlerini yerine getirir. Bu butonların kullanılabilmesi için tablonun herhangi bir kaydının seçilmiş olması gerekir. Butonlar seçilen satırdaki bilgiler üzerinde işlem yapar.



Gelişmiş Arama Ekranı : Bu ekranda tüm listeler içinde arama yapılabilir. Arama kriterleri öncelik, durum ve tarih aralığı olarak seçilebilir. Bir önceki ekranda bulunan butonlar aynı işlevi yapar. Listele butonu seçilen kriterlere göre görevleri listeler.



Liste Düzenleme Ekranı : Bu ekranda, listelerim ekranındaki listelerin adları değiştirilebilir. textBox içine yeni liste adını yazdıktan sonra kaydet butonunun tıklanması üzerine değişiklikler kaydedilir. Program bir sonraki açılışta bu bilgileri veri tabanından çekeceği için veriler kaybolmaz.



Son Hareketler Ekranı : Bu ekranda son tamamlanan işler listelenir. Liste ekranı ve gelişmiş arama ekranındaki seçilen görevi tamamla butonu, tabloda seçilen görevin durumunu tamamlandı olarak işaretlerken aynı zamanda buradaki tabloya kayıt ekler.

Zorlandığım Kısımlar

SQLite Veritabanında Tarih Sorunları : Programı yaparken en fazla zamanımı alan sorun bu oldu. SQLite veritabanına insert ettiğim date bilgisini, select ile çekip adapter ile dataset içerisine fill ederken hata mesajı aldım. Datetimepicker nesnesinin textini ve value bilgisini almayı denedim. Hiçbir yöntem işe yaramadı. Sonunda stack overflow 'da bu konu hakkında yazılan yazıları okudum ve çözüme ulaştım. Veritabanına aktaracağım datetimepicker.value değerine özel format ile yyyy-mm-dd şeklinde dönüşüm uygulamam sorunun çözülmesini sağladı.

Nesnelere Erişim : Programda çok fazla user control kullanmamdan dolayı controllerin birbirleri arasında nesne komutları göndermesi gerekiyor. Bütün bunları form içinde form açarak yaptım. Yani A formundan B formundaki nesne için komut yazacaksam, A formu içinde B formu tanımlayıp bu form üzerinden işlem yaptım. Sorunlarımı çözdü.

Kullandığım Metotlar

Program içerisinde çoğunlukla veri tabanı komutları kullandığım için DBCommands adlı bir class içerisinde bu komutları metot olarak açtım. Projede çok fazla metot kullandım. En önemlilerinden bahsedeceğim.

`dbCreate();`

Programın en önemli metodudur. Program açıldığında hedef dosya yolunda veri tabanı dosyası yoksa yeni bir veri tabanı açar. Açtığı veri tabanına gerekli tabloları açar ve gerekli bilgiler ile doldurur.

```
public static void dbCreate()
{
    if (!File.Exists("taskup.db"))
    {
        SQLiteConnection.CreateFile("taskup.db");
        dbConnect();
        using (con)
        {
            using (SQLiteCommand cmd = new SQLiteCommand(con))
            {
                cmd.CommandText = @"Create table if not exists lists (
                    listno INTEGER PRIMARY KEY,
                    listname nvarchar(15))";
                cmd.ExecuteNonQuery();
                dbDisconnect();
            }
        }
        dbConnect();
        using (con)
        {
            using (SQLiteCommand cmd2 = new SQLiteCommand(con))
            {
                cmd2.CommandText = @"Create table if not exists tasks (
                    taskno INTEGER PRIMARY KEY AUTOINCREMENT,
                    listno INTEGER,
                    date DATE,
                    title nvarchar(20),
                    text TEXT,
                    priority nvarchar(10),
                    info nvarchar(15))";
                cmd2.ExecuteNonQuery();
                dbDisconnect();
            }
        }
        dbConnect();
        using (con)
        {
            using (SQLiteCommand cmd3 = new SQLiteCommand(con))
            {
                cmd3.CommandText = @"Create table if not exists history (
                    processingno INTEGER PRIMARY KEY AUTOINCREMENT,
                    text TEXT)";
                cmd3.ExecuteNonQuery();
                dbDisconnect();
            }
        }
        dbConnect();
        using (con)
        {
            using (SQLiteCommand cmd = new SQLiteCommand(con))
            {
                cmd.CommandText = "insert into lists (listno,listname) values";
                cmd.ExecuteNonQuery();
                cmd.CommandText = "insert into lists (listno,listname) values";
                cmd.ExecuteNonQuery();
                cmd.CommandText = "insert into lists (listno,listname) values";
                cmd.ExecuteNonQuery();
                cmd.CommandText = "insert into lists (listno,listname) values";
                cmd.ExecuteNonQuery();
                dbDisconnect();
            }
        }
    }
}
```

```

public static DataSet getList(string a)
{
    con = new SQLiteConnection("Data Source=taskup.db;Version=3;");
    SQLiteDataAdapter da = new SQLiteDataAdapter("Select *From "+a, con);
    DataSet ds = new DataSet();
    con.Open();
    da.Fill(ds, "Lists");
    con.Close();
    return ds;
}

public static DataSet getList(string a, string b, string c)
{
    con = new SQLiteConnection("Data Source=taskup.db;Version=3;");
    SQLiteDataAdapter da = new SQLiteDataAdapter("Select TaskNo,Date,Title,Text,Priority,Info From " + a + " Where " + b + "=" + c, con);
    DataSet ds = new DataSet();
    con.Open();
    da.Fill(ds, "Lists");
    con.Close();
    return ds;
}

```

getList();

Bu metotta overloading kullandım. Metoda tek parametre gönderirseniz gönderdiğiniz parametreye göre tablo bilgisini dataset türünde return ediyor. Üç parametre gönderecek olursanız belirlediğiniz tablodaki belirlediğiniz koşuldaki verileri dataset türünde return ediyor.

dbConnect(); ve dbDisconnect();

Programın her parçasında kullandığım iki metottur. İsminden de anlaşıldığı gibi veri tabanı bağlantısını açar ve kapatır. con.Close(); yerine dbDisconnect(); fonksiyonunu yazmamın nedeni, kodu yazarken farklı terimlerin kafamı karıştırmaması içindir. Değişken ve metot isimlerini yaptığı işe göre belirlemek kolaylık sağlar.

```

public static void compTask(string a)
{
    dbConnect();
    using (con)
    {
        using (SQLiteCommand cmd = new SQLiteCommand(con))
        {
            cmd.CommandText = "update Tasks set Info='Tamamlandı' Where TaskNo=" + a;
            cmd.ExecuteNonQuery();
            dbDisconnect();
        }
    }
}

```

compTask();

Bu metot, gönderdiğiniz parametreye göre veri tabanındaki görev kaydını tamamlandı olarak güncelliyor.

```

public static void updateTask(String p1, string p2, string p3, string p4, int p5)
{
    dbConnect();
    using (con)
    {
        using (SQLiteCommand cmd = new SQLiteCommand(con))
        {
            cmd.CommandText = "update Tasks set Date='"+p1+"', Title='"+p2+"', Text='"+p3+"', Priority='"+p4+"' where TaskNo=" + p5.ToString();
            cmd.ExecuteNonQuery();
            dbDisconnect();
        }
    }
}

```

updateTask();

Bu metot, gönderdiğiniz parametrelere göre veri tabanındaki görev kaydını güncelliyor.

```

public static void insertTask(string p1, DateTime p2, string p3, string p4, string p5, string p6)
{
    dbConnect();
    using (con)
    {
        using (SQLiteCommand cmd = new SQLiteCommand(con))
        {
            string date = p2.ToString("yyyy-MM-dd");
            cmd.CommandText = "insert into Tasks (ListNo,Date,Title,Text,Priority,Info) values (@p1,@p2,@p3,@p4,@p5,@p6)";
            cmd.Parameters.AddWithValue("@p1", p1);
            cmd.Parameters.AddWithValue("@p2", date);
            cmd.Parameters.AddWithValue("@p3", p3);
            cmd.Parameters.AddWithValue("@p4", p4);
            cmd.Parameters.AddWithValue("@p5", p5);
            cmd.Parameters.AddWithValue("@p6", p6);
            cmd.ExecuteNonQuery();
            dbDisconnect();
        }
    }
}

```

insertTask();

Bu metot, gönderdiğiniz parametreye göre veri tabanına yeni görev ekliyor.

```

public static void deleteTask(string a)
{
    dbConnect();
    using (con)
    {
        using (SQLiteCommand cmd = new SQLiteCommand(con))
        {
            cmd.CommandText = "delete from Tasks Where TaskNo=" + a;
            cmd.ExecuteNonQuery();
            dbDisconnect();
        }
    }
}

```

deleteTask();

Bu metot, gönderdiğiniz parametreye göre veri tabanındaki görev bilgilerini siliyor.

```

public static void historyWrite(int a)
{
    dbConnect();
    using (con)
    {
        using (SQLiteCommand cmd = new SQLiteCommand(con))
        {
            cmd.CommandText = "select Title from Tasks where TaskNo="+a;
            string title = cmd.ExecuteScalar().ToString();
            cmd.CommandText = "insert into History (Text) values ('"+title+" başlıklı görevinizi tamamladınız.')";
            cmd.ExecuteNonQuery();

            dbDisconnect();
        }
    }
}

```

historyWrite();

Bu metot, görev tamamlama esnasında göreve ait başlık bilgisini, son işlemler tablosuna aktarıyor. Son hareketler ekranında bu metodun eklediği kayıtlar görünüyor.

```

public static void clickListeler()
{
    showMenu(Taskup.pmenuListeler);
    Taskup.pcolor_bar.BackColor = Color.Magenta;
    Taskup.pbtnListeler.BackColor = Color.Purple;
    Taskup.pbtnAyarlar.BackColor = Color.Transparent;
    Taskup.pbtnHareketler.BackColor = Color.Transparent;
    Taskup.pcolor_bar.Location = new Point(Taskup.pcolor_bar.Location.X, Taskup.pbtnListeler.Location.Y);
}

public static void clickBasarimlar()
{
    showMenu(Taskup.pmenuHareketler);
    Taskup.pcolor_bar.BackColor = Color.Yellow;
    Taskup.pbtnHareketler.BackColor = Color.DarkOrange;
    Taskup.pbtnListeler.BackColor = Color.Transparent;
    Taskup.pbtnAyarlar.BackColor = Color.Transparent;
    Taskup.pcolor_bar.Location = new Point(Taskup.pcolor_bar.Location.X, Taskup.pbtnHareketler.Location.Y);
}

public static void clickAyarlar()
{
    showMenu(Taskup.pmenuAyarlar);
    Taskup.pcolor_bar.BackColor = Color.Blue;
    Taskup.pbtnAyarlar.BackColor = Color.SkyBlue;
    Taskup.pbtnListeler.BackColor = Color.Transparent;
    Taskup.pbtnHareketler.BackColor = Color.Transparent;
    Taskup.pcolor_bar.Location = new Point(Taskup.pcolor_bar.Location.X, Taskup.pbtnAyarlar.Location.Y);
}

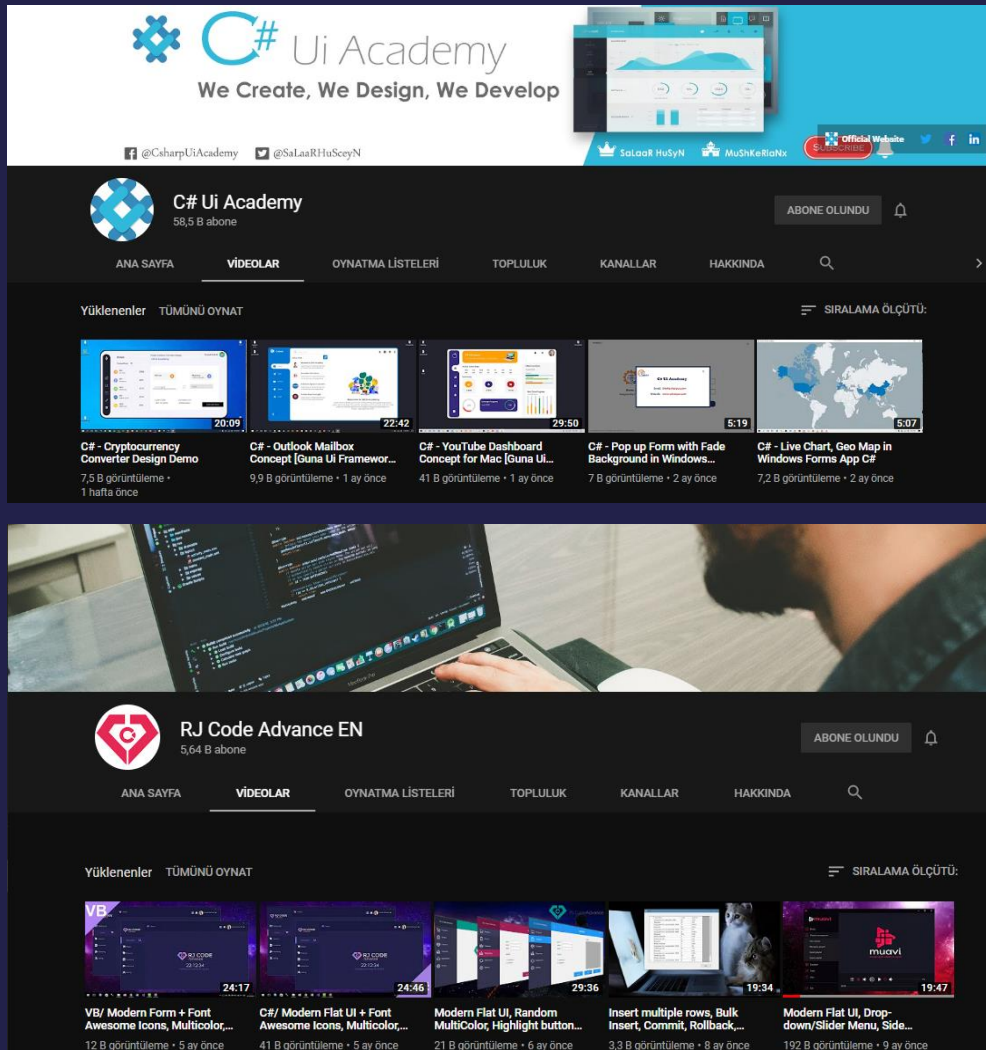
```

clickListeler(); - clickBasarimlar(); - clickAyarlar();

Son olarak bu metotlar da ana ekrandaki butonların açılır menü gibi görsel işlevlerini yapmasına yarıyor.

Tasarım

Programı yaparken en çok zevk aldığım bölüm tasarım bölümü. Bu programda ilk defa nesnelerin dock özelliğini kullandım. Daha önceki yaptığım programlardan daha iyi bir görsel ortaya çıktı. Tasarımı yaparken 'RJ Code Advance' ve 'C# UI Academy' adlı iki youtube kanalından yararlandım. Bu kanallar uygulamalı olarak Visual Studio üzerinden form tasarımı yapıyorlar. İzlemekte uygulamak ta çok eğlenceli.



Kaynaklar

Uygulama Tasarımı

[RJ Code](#)

[C# UI Academy](#)

SQLite Kullanımları

[Turkcell](#)

[Algoritma Uzmanı](#)

[W3School](#)

Hata Çözümleri

[Stack Overflow](#)

[Microsoft](#)