**T.C.  
  
CELAL BAYAR ÜNVERSİTESİ  
  
KIRKAĞAÇ MESLEK YÜKSEKOKULU  
  
BİLGİSAYAR TEKNOLOJİLERİ VE PROGRAMLAMA**

****

**Ders Programı Hazırlama Otomasyonu**

Hazırlayanlar

**221809028 Ömer DÖNER**

**221809044 Muhammet Raşit KOÇAK**

**221809075 Yasin BALCI**

Danışman

**Öğr.Gör. Murat ALBAYRAK**

***İçindekiler Tablosu***

[***Giriş*** 3](#_Toc158204956)

[***Uygulamayı kurma:*** 3](#_Toc158204957)

[***Visual Studio Kurulumu:*** 3](#_Toc158204958)

[***Uygulamayı Açma:*** 3](#_Toc158204959)

[***Proje Açma:*** 3](#_Toc158204960)

[***SQL Server Ayarlama:*** 3](#_Toc158204961)

[***Uygulama ne işe yarıyor ?*** 4](#_Toc158204962)

[***Hızlı ve Verimli Program Oluşturma:*** 5](#_Toc158204963)

[***Çakışmaları Önleme:*** 5](#_Toc158204964)

[***En Uygun Atamaları Yapma:*** 5](#_Toc158204965)

[***Esneklik ve Uyum:*** 5](#_Toc158204966)

[***Teklif edilen Sistem*** 5](#_Toc158204967)

[***1. Kısıtlı Yetkili Kullanıcı*** 5](#_Toc158204968)

[***2. Tam Yetkili Kullanıcı*** 6](#_Toc158204969)

[***Otomasyonumuz*** 6](#_Toc158204970)

[***Kullanılabilirlik*** 6](#_Toc158204971)

[***Güvenilirlik*** 7](#_Toc158204972)

[***Performans*** 7](#_Toc158204973)

[***Desteklenebilirlik*** 7](#_Toc158204974)

[***İmplementasyon*** 7](#_Toc158204975)

[***Arayüz*** 7](#_Toc158204976)

[***Gizlilik Gereksinimi*** 7](#_Toc158204977)

[***Aktörler*** 7](#_Toc158204978)

[***Olaylar*** 8](#_Toc158204979)

[***Senaryolar*** 8](#_Toc158204980)

[***Veri Tabanı Diyagramı:*** 10](#_Toc158204981)

[***Program Aşamaları Kodlarla*** 11](#_Toc158204982)

[***Çalıştırma:*** 11](#_Toc158204983)

[***Giriş Yapma:*** 11](#_Toc158204984)

[***Menü Oluşturma :*** 12](#_Toc158204985)

[Veriyi Sil: 20](#_Toc158204986)

[Veriyi Kayıt Et: 21](#_Toc158204987)

[***Veriyi Güncelle:*** 21](#_Toc158204988)

[***Cıktı Al:*** 21](#_Toc158204989)

[***Algoritma Aşamaları Kodlarla*** 23](#_Toc158204990)

[***1. Öğretmenlerin Müsaitliği*** 23](#_Toc158204991)

[***4. Ders Bilgileri*** 24](#_Toc158204992)

[Alogirtma Akış Diyagramı… 26](#_Toc158204993)

# ***Giriş***

## ***Uygulamayı kurma:***

### ***Visual Studio Kurulumu:***

* Öncelikle Visual Studio nun resmi web sitesinden uygun sürümü indirin ve kurulumu gerçekleştirin.
* Kurulum sihirbazını takip ederek temel ayarları seçip, Visual Studio nun bilgisayarınıza başarılı bir şekilde kurulmasını sağlayın.

### ***Uygulamayı Açma:***

* Kurulumun tamamlanmasının ardından Visual Studio uygulamasını başlatın.
* Sol üst köşede bulunan "File" menüsüne tıklayın.

### ***Proje Açma:***

* "File" menüsünden "Open" seçeneğine tıklayın ve bilgisayarınızda bulunan proje dosyanızı seçin.

## ***SQL Server Ayarlama:***

Otomasyonu sorunsuz bir şekilde kullanmak ve veritabanı işlemlerini gerçekleştirmek için SQL Server bağlantı bilgilerinizi doğru bir şekilde ayarlamak oldukça önemlidir. Aşağıda, bağlantı bilgilerini düzenlemek için adım adım bir rehber bulabilirsiniz:

Visual Studio yu açtıktan sonra ara kısmına sunucu gezgini yazın tıklayın

metin, ekran görüntüsü, yazı tipi, tasarım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu şekilde bir ekran olucak karşınızda x ifadesinin sağındaki butona tıklayın

metin, elektronik donanım, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

Veri kaynağını seçip devam et deyin

metin, ekran görüntüsü, sayı, numara, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

Sunucu adınızı girin sonra tamam a tıklayın veri tabanı bağlantısı tamamlandı.

## ***Uygulama ne işe yarıyor ?***

Modern eğitim ortamlarında yöneticilerin en büyük zorluklarından biri, öğretmen ve ders kaynaklarını etkili bir şekilde yöneterek, öğrencilere en iyi eğitimi sunmaktır. Bu noktada, ders programı oluşturmak sıklıkla zaman alıcı ve karmaşık bir süreç olabilir. Ancak, şimdi bu zorluğu ortadan kaldıran bir çözüm var: Akıllı Ders Programı Otomasyonu!

Bu benzersiz uygulama, yöneticilere öğretmen bilgileri, ders içerikleri, okul zamanı ve bölüm detayları gibi kritik verileri girdikten sonra, ders çakışmalarını önleyen ve en uygun şekilde bölümlere atama yapabilen bir akıllı sistem sunuyor. İşte bu uygulamanın sunduğu faydalar:

### ***Hızlı ve Verimli Program Oluşturma:***

Geleneksel yöntemlerle saatler süren ders programı oluşturma sürecini, bu otomasyon sayesinde sadece birkaç tıklama ile tamamlayabilirsiniz. Bu, yöneticilere zaman kazandırarak daha önemli görevlere odaklanma imkanı sunar.

### ***Çakışmaları Önleme:***

Uygulama, dersler arasındaki çakışmaları etkili bir şekilde engeller. Bu, öğrencilere daha düzenli bir program sunmanın yanı sıra, öğretmenlere de daha etkili bir şekilde ders vermelerine olanak tanır.

### ***En Uygun Atamaları Yapma:***

Dersleri, öğretmenlerin uzmanlık alanlarına ve bölümlere en uygun şekilde atayarak, öğrencilerin daha iyi bir eğitim almasını sağlar. Bu sayede, her öğrenci kendi ilgi ve yeteneklerine uygun derslere katılabilir.

### ***Esneklik ve Uyum:***

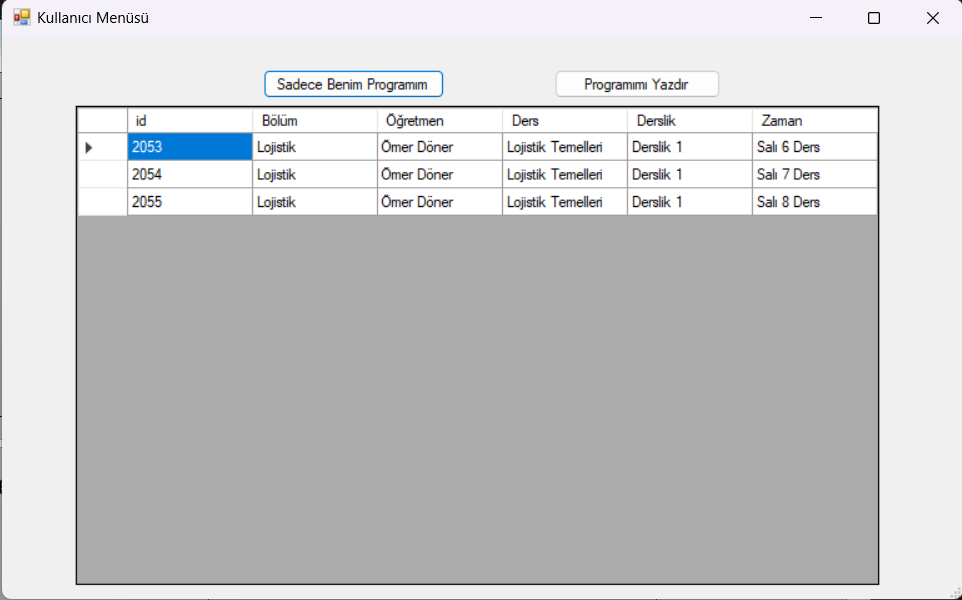
Uygulama, değişen okul koşullarına ve öğretmen programlarındaki güncellemelere hızlı bir şekilde adapte olabilir. Bu da yöneticilere esneklik sağlayarak, herhangi bir aksaklığa karşı hazırlıklı olmalarını sağlar.

Bu Akıllı Ders Programı Otomasyonu ile, eğitim kurumları daha hızlı, daha düzenli ve daha verimli bir şekilde ders programları oluşturabilir. Bu da öğrencilerin ve öğretmenlerin en iyi şekilde faydalanmasını sağlayarak eğitim kalitesini artırır. Zamanınızı daha etkili bir şekilde yönetmek ve daha iyisini başarmak için şimdi bu benzersiz uygulamayı keşfedin!"

## ***Teklif edilen Sistem***

Sistemde 2 Adet kullanıcı vardir. Tam Yetkili ve Kısıtlı yetkili olmak üzere şimdi detaylıca anlatalım..

### ***1. Kısıtlı Yetkili Kullanıcı***

Kısıtlı yetkili kullanıcılar öğretmenlerdir ve öğretmenler programda sadece kendi bilgilerini görebilmektedir. Ayrıca cıktı olarak hem tablo halinde hemde ders programı halinde de cıktı alabilirler…  
  
  
Kısıtlı yetkili kullanıcılar örneğin bu şekilde verilerini görebilirler.

### ***2. Tam Yetkili Kullanıcı***

Tam yetkili kullanıcılar menülerde herşey yapabilecek kullanıcılardır. Ufak bir ekran görüntüsünden neler yapabileceklerini gösterelim…

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

Tam Yetkili bir kullanıcı buradaki tüm bilgileri eksiksiz bir şekilde girdiği zaman ders programını hazırlayabilir. Eksik bilgilerde hata uyarı mesajlarımız mevcuttur.

## ***Otomasyonumuz***

### ***Kullanılabilirlik***

Projede, ders programı hazırlama sürecini hızlandırmak için kullanıcıların ihtiyaç duyduğu ana işlevlere hızlı erişim sağlayan menüler bulunmalıdır. Bu menüler, kullanıcıların istedikleri işlemlere kolayca ulaşmalarını sağlamalı ve zaman kaybını minimuma indirmelidir. Oluşturulan program istenildiği gibi cıktı alabilir ve basit bir arayüzü sayesinde kullanıcılara güçlü zorluklar yaratmamaktadır. Ayrıca en üst düzeyde optimize sağlanmış ve geliştirilebilmesi çok basit bir şekilde kodlanmıştır

### ***Güvenilirlik***

Ders programı otomasyonunda verilerin korunması ve güvenliği oldukça önemlidir. Yaşanılabilecek 3. Parti uygulamala açıklarının ve illegal girişleri engellemek için ufak güvenilir kod bütünlüğü kullanarak bu açıkları kapattık. Mesela login sistemindeki sql saldırılarını engelledik. Yada DDOS saldırılarını azaltmak için programda ufak engeller ekledik. Onun dışında database’nizi kodumuzun içerisinde güvenilir bir şekilde gömdüğümüz için bizlere güvenebilirsiniz. Ama en önemlisi yaşayacağınız en ufak sorunda yanınızda bulunacağımızı bilmelisiniz.

### ***Performans***

Performans olaraktan verileri sürekli databaseden çekip yaşayacağımız ağ trafiğinden dolayı verilerimizi nodelere kayıt edip en son işlemleri veritabanından yapıyoruz bu sayede daha az dişardan bağlantı cekerek performansımızı güncellendirdik.

### ***Desteklenebilirlik***

Projede kullanılan veritabanının transaction ve foreign key desteğinin olması gerekli. Programın kurulduğu platformun da C# destekli olmalı ve C# programlarını çalıştırabilmeli. Özgür yazılım lisansı sayesinde yasal gereksinimleri bulunmamakta

### ***İmplementasyon***

Sistem için verilen veritabanlarını dikkatli bir şekilde bağlaştırıp UML Diyagramımızı oluşturduk aşağıda diyagramımız mevcut ve buna göre gerekli c# kodlarında bağlantılarımızı destekledik.

### ***Arayüz***

Otomasyonda kullanıcı arabirimi pencerelerdir. Tek bir pencerede yer alacak programda, kullanıcıyı login sayfası karşılar. Login kullanıcı, rolüne göre kendine ait bilgilere ulaşır. Görsel özellikler için gereksinim dökümanı ayrıca hazırlanacaktır. Kullanıcı düğmelere basarak veya formları doldurarak işlem yapacaktır. Ara yüzün çıktısı ekran, girdileri ise Mouse ve klavyedir.

### ***Gizlilik Gereksinimi***

Kullanıcıların sınırlandırılması, sadece kendilerine ait bölümleri görebilmesi ve gerektiğinde bu ayarların değiştirilmesi yüksek derecede önemli. Veritabanında, dataların güvenliği için ek bir özellik bulunmuyor. Sunucunun başına geçen veya dosyaları çalan birisi kayıtlara ulaşabilir. Sunucu güvenliği işletim sisteminin kendi güvenliği ile eş. Program açılırken kullanıcı adı ve şifre soracak. Sistem bu bilgileri kullanarak izinlerini kontrol edecektir. Kullanıcılar veritabanına başka bir programla bağlanıp bilgileri görebilir. Programda kullanılan gizlilik sadece verinin bütünlüğü ve iş bölümü için. İstenmeyen kullanıcıların sisteme bağlanmasını engellemek işletim sisteminin sorumluluğundadır.

### ***Aktörler***

|  |  |
| --- | --- |
| **Aktör** | **Tanım** |
| Tam Yetkili | Sistem içerisinde tamamen bütün yetkinin ait olduğu ve sistem içerisindeki bütün düzenlemeleri yapan kullanıcıdır. Kısaca sistemdeki YÖNETİCİ’dir |
| Kısıtlı Yetkili | Sadece ders programını görüntüler ve sistemde cıktı alabilir. Kısaca yetkisi olmayan Öğretmenlerdir. |
| Sistem | Ders programı oluşturma, çakışmaları kontrol etme, otomatik derslik ve laboratuvar atama gibi görevleri gerçekleştiren bir sistem aktörüdür.  Çakışmaları kontrol eder ve yöneticiye uyarılar verir, ancak müdahale etmez. Yönetici, uyarıları değerlendirip istediği şekilde değişiklik yapabilir. |

### ***Olaylar***

Sistemin haftalık ders programı hazırlama otomasyonu olaylarını şu şekilde sıralayabiliriz:

**Hoca Bilgisi Girişi:**

Yönetici, hocaların bilgilerini (unvanları ile birlikte) sisteme girebilir.

**Ders Bilgisi Girişi:**

Yönetici, derslerin bilgilerini (ders kodu, adı, ders saati) sisteme girebilir.

**Hoca-Ders Atama:**

Kullanıcı, hangi hoca hangi derse gireceğini sisteme girebilir.

Bir hoca birden fazla derse atanabilir.

Bir derse birden fazla hoca atanabilir.

**Şube ve Program Ayarları:**

Yönetici, bir şube için günlük ders saatlerini, başlangıç saatini, ders süresini, teneffüs süresini, öğle tatili süresini belirleyebilir.

**Çakışma Kontrolü:**

Sistem, aynı gün ve saatte ismi farklı birden fazla derse atanmış bir hocanın durumunu kontrol eder.

Eğer çakışma varsa, yöneticiye uyarı verir.

**Müsaitlik Girişi:**

Hocalar, müsait oldukları günleri ve saatleri sisteme girebilir.

**Derslik ve Laboratuvar Atama:**

Sistem, bir dersin laboratuvarda yapılması gerekiyorsa otomatik olarak laboratuvar ataması yapar.

Eğer müsait bir laboratuvar bulunamıyorsa, sistem yöneticiye müsait derslikleri önerir.

**Otomatik Derslik Atama:**

Sistem, bir dersin öğrenci sayısına göre en uygun dersliğe otomatik olarak atama yapar, en az boş yer bırakan dersliği seçer.

**Çakışma İzni:**

Yönetici, aynı gün ve saatte ismi farklı birden fazla derse atanmış bir hocanın durumu hakkında uyarı aldığında, sistemin önerdiği çözümü değerlendirip onay verebilir.

**Program Çıktıları:**

Kullanıcı, bölüm, sistem ve hoca bazında ayrı ayrı haftalık program çıktıları alabilir.

**Manuel Değişiklikler:**

Yönetici, ders programında istediği şekilde manuel değişiklikler yapabilir.

Sistem, kriterlere uygun olmayan durumlar hakkında yöneticiyi uyarır, ancak müdahale etmez.

### ***Senaryolar***

***Hoca Eklemek:***

Yönetici, yeni bir hoca ekleyerek unvan, ad, soyad gibi bilgileri sisteme kaydeder.

***Ders Eklemek:***

Yönetici, yeni bir ders ekleyerek ders kodu, ders adı, ders saati gibi bilgileri sisteme kaydeder.

***Hoca-Ders Ataması Yapmak:***

Kullanıcı, belirli bir hoca için hangi dersin atanacağını seçer ve sisteme kaydeder.

***Şube Programı Ayarlamak:***

Yönetici, bir şube için günlük ders saatleri, başlangıç saati, teneffüs süresi gibi program ayarlarını düzenler.

***Çakışma Kontrolü:***

Sistem, bir hoca için çakışan ders atamalarını kontrol eder ve yöneticiye uyarı verir.

***Müsaitlik Bilgisi Girmek:***

Hoca, kendi müsait olduğu günleri ve saatleri sisteme girer.

***Derslik ve Laboratuvar Ataması:***

Sistem, bir dersin laboratuvarda yapılması gerekiyorsa otomatik olarak laboratuvar ataması yapar.

***Otomatik Derslik Atama:***

Sistem, derslerin öğrenci sayılarına göre en uygun dersliklere otomatik olarak atama yapar.

***Çakışma İzni Verme:***

Yönetici, çakışan ders atamaları konusunda sistem tarafından verilen uyarılara izin verir.

***Program Çıktısı Almak (Bölüm):***

Kullanıcı, belirli bir bölüm için haftalık ders programını sisteme sorgular ve çıktı alır.

***Program Çıktısı Almak (Sistem):***

Kullanıcı, genel sistem haftalık ders programını sorgular ve çıktı alır.

***Program Çıktısı Almak (Hoca):***

Kullanıcı, belirli bir hoca için haftalık ders programını sisteme sorgular ve çıktı alır.

***Manuel Değişiklik Yapmak:***

Yönetici, ders programında manuel değişiklik yapar ve sistem tarafından verilen uyarıları değerlendirir.

***Ders Sayısı Azaltmak:***

Yönetici, bir dersin haftalık saat sayısını azaltır ve sistemden olası çakışma uyarılarını değerlendirir.

***Derslik Kapasitesini Güncellemek:***

Yönetici, bir dersliğin kapasitesini günceller ve sistemin otomatik atama algoritmasını etkiler.

***Hoca Bilgisi Güncellemek:***

Yönetici, bir hocanın bilgilerini günceller ve sistemdeki ilgili atamalara yansıtır.

***Ders Bilgisi Güncellemek:***

Yönetici, bir dersin bilgilerini günceller ve sistemdeki ilgili atamalara yansıtır.

***Derslik Bilgisi Eklemek:***

Yönetici, yeni bir derslik ekler ve sistemde otomatik atama algoritmasını etkiler.

***Tüm Ders Atamalarını Silmek:***

Yönetici, tüm ders atamalarını siler ve sistemde temiz bir başlangıç yapar.

***Çakışma Kontrolsüz İzin:***

Yönetici, çakışan ders atamalarına sistem uyarısı olmadan izin verir.

## ***Veri Tabanı Diyagramı:***

Veritabanında kurduğumuz bağlantılar aşağıda bir görselde belirtilmiştir. Belirtilen görselde kurulan bağlantılar (FOREIGN KEY) güncelleme işlemlerinde veri güncellemesi için ayrıntılı kullanılmıştır.

metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

# ***Program Aşamaları Kodlarla***

## ***Çalıştırma:***

metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

Program ilk olarak bu class’tan başlamaktadır.

## ***Giriş Yapma:***

Uygulamayı çalıştıktan sonra login ekranına atar login ekranında ise kullanıcının olup olmamasını tabloda o kullanıcı adına ve şifresine sahip 1 kullanıcı varmı yokmu diye bir sorgu yapıp kullanıcının doğruluğunu aldım.

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturulduBu kod parçasında da gördüğünüz üzere…

## ***Menü Oluşturma :***

Asıl konumuzdan birisi menü oluşturma, menü oluşumlarını en basit olacak şekilde ayarlamaya çalıştım. Şimdi ufak bir kod parçasından ilerleyerek nasıl oluştuğuna bakalım…

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu  
  
yazılım, multimedya yazılımı, bilgisayar simgesi, metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

## 

namespace dersProgrami.Manager

{

public partial class teacher : Form

{

public teacher()

{

InitializeComponent();

}

int SecilenOgrId;

int SecilenDersId;

DataSet1TableAdapters.ogretmenTableAdapter dtogr = new DataSet1TableAdapters.ogretmenTableAdapter();

DataSet1TableAdapters.ogretmenatamasıTableAdapter dtogratama=new DataSet1TableAdapters.ogretmenatamasıTableAdapter();

DataSet1TableAdapters.derslerTableAdapter dtdersler = new DataSet1TableAdapters.derslerTableAdapter();

DataSet1TableAdapters.OgrAtamaJoinTableAdapter dtOgrAtamaJoin = new DataSet1TableAdapters.OgrAtamaJoinTableAdapter();

DataSet1TableAdapters.OgretmenMusaitlikTableAdapter dtogrMusaitlik = new DataSet1TableAdapters.OgretmenMusaitlikTableAdapter();

DataSet1TableAdapters.OgrMusaitlikJoinTableAdapter dtOgrMusaitlikJoin = new DataSet1TableAdapters.OgrMusaitlikJoinTableAdapter();

private void OgretmenListele()

{

dataGridView2.DataSource= dtogr.GetOgretmen();

dataGridView3.DataSource = dtOgrAtamaJoin.GetOgrAtamaJoin();

dataGridView1.DataSource = dtOgrMusaitlikJoin.GetOgretmenMusaitlikJoin();

cmbx\_ogrmusait.DataSource = dtogr.GetOgretmen();

cmbx\_ogrmusait.DisplayMember = "ad";

cmbx\_ogrmusait.ValueMember = "id";

cmbx\_ogrAtama.DataSource = dtogr.GetOgretmen();

cmbx\_ogrAtama.DisplayMember = "ad";

cmbx\_ogrAtama.ValueMember = "id";

}

private void teacher\_Load(object sender, EventArgs e)

{

OgretmenListele();

OgretmenAtamaListesi();

OgretmenMusaitlikListesi();

cmbx\_ogrAtama.DataSource = dtogr.GetOgretmen();

cmbx\_ogrAtama.DisplayMember = "ad";

cmbx\_ogrAtama.ValueMember = "id";

cmbx\_OgrDersAtama.DataSource = dtdersler.GetDersler();

cmbx\_OgrDersAtama.DisplayMember = "ders";

cmbx\_OgrDersAtama.ValueMember = "id";

cmbx\_ogrmusait.DataSource = dtogr.GetOgretmen();

cmbx\_ogrmusait.DisplayMember = "ad";

cmbx\_ogrmusait.ValueMember = "id";

}

private void btn\_ogr\_kaydet\_Click(object sender, EventArgs e)

{

try

{

dtogr.InsertOgretmen(cbmx\_yetki.Text, cmbx\_ünvan.Text, veri\_tc.Text, veri\_telefon.Text, veri\_ad.Text + " " + veri\_soyad.Text, veri\_sifre.Text);

MessageBox.Show("Ogretmen sisteme eklendi", "bilgi", MessageBoxButtons.OK, MessageBoxIcon.Information);

OgretmenListele();

}

catch (System.Data.SqlClient.SqlException ex)

{

int errorCode = ex.Number; // SQL hatasının numarasını al

if (errorCode == 547)

{

MessageBox.Show("Hata: Lütfen tüm değerleri minimum 3 harf ile girin.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (errorCode == 2627)

{

string errorMessage = ex.Message.ToLower();

if (errorMessage.Contains("telefon"))

{

MessageBox.Show("Hata: Telefon numarası benzersiz olmalıdır.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (errorMessage.Contains("tc"))

{

MessageBox.Show("Hata: TC kimlik numarası benzersiz olmalıdır.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

MessageBox.Show("Bilinmeyen bir hata oluştu. Hata kodu: " + errorCode.ToString(), "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

else

{

MessageBox.Show("Bilinmeyen bir hata oluştu. Hata kodu: " + errorCode.ToString(), "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void btn\_ogr\_sil\_Click(object sender, EventArgs e)

{

if (int.TryParse(txt\_ogr\_id.Text, out int ogretmenId))

{

dtogr.DeleteOgretmen(ogretmenId);

MessageBox.Show("Öğretmen " + ogretmenId + " silindi", "Bilgi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

OgretmenListele();

}

else

{

MessageBox.Show("Geçersiz Öğretmen ID", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void btn\_ogr\_guncelle\_Click(object sender, EventArgs e)

{

try

{

if (int.TryParse(txt\_ogr\_id.Text, out int ogretmenId))

{

dtogr.UpdateOgretmen(cbmx\_yetki.Text, cmbx\_ünvan.Text, veri\_tc.Text, veri\_telefon.Text, veri\_ad.Text + " " + veri\_soyad.Text , veri\_sifre.Text, ogretmenId);

MessageBox.Show("Öğretmen " + ogretmenId + " güncellendi", "Bilgi", MessageBoxButtons.OK, MessageBoxIcon.Information);

OgretmenListele();

}

else

{

MessageBox.Show("Geçersiz Öğretmen ID", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

catch (System.Data.SqlClient.SqlException ex)

{

int errorCode = ex.Number; // SQL hatasının numarasını al

if (errorCode == 547)

{

MessageBox.Show("Hata: Lütfen tüm değerleri minimum 3 harf ile girin.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (errorCode == 2627)

{

MessageBox.Show("Hata: Telefon ve TC lütfen benzersiz yapın.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

MessageBox.Show("Bilinmeyen bir hata oluştu. Hata kodu: " + errorCode.ToString(), "Hata" + ex.Number, MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void btn\_ogr\_ogr\_atama\_Click(object sender, EventArgs e)

{

if (cmbx\_ogrAtama.SelectedItem != null && cmbx\_OgrDersAtama.SelectedItem != null)

{

SecilenDersId = Convert.ToInt32(cmbx\_OgrDersAtama.SelectedValue);

SecilenOgrId = Convert.ToInt32(cmbx\_ogrAtama.SelectedValue);

}

dtogratama.UpdateOgrAtama(SecilenOgrId,SecilenDersId,int.Parse(txt\_ogrAtma\_id.Text));

MessageBox.Show("Ogretmen ataması" + txt\_ogr\_id.Text + " güncellendi", "bilgi", MessageBoxButtons.OK, MessageBoxIcon.Information);

OgretmenAtamaListesi();

}

private void btn\_kaydet\_atama\_Click(object sender, EventArgs e)

{

try

{

SecilenDersId = Convert.ToInt32(cmbx\_OgrDersAtama.SelectedValue);

SecilenOgrId = Convert.ToInt32(cmbx\_ogrAtama.SelectedValue);

dtogratama.InsertOgrAtama(SecilenOgrId, SecilenDersId);

MessageBox.Show("Öğretmen ataması yapıldı", "Bilgi", MessageBoxButtons.OK, MessageBoxIcon.Information);

OgretmenAtamaListesi();

}

catch (System.Data.SqlClient.SqlException ex) {

if (ex.Number == 2627)

{

MessageBox.Show("Bu atama zaten yapıldı.", "Hata " + ex.Number, MessageBoxButtons.OK, MessageBoxIcon.Error);

} else if (ex.Number == 547)

{

MessageBox.Show("Lütfen boş bırakmayın.", "Hata " + ex.Number, MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

MessageBox.Show("Hata ", "Hata " + ex.Number, MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void btn\_sil\_ogr\_atama\_Click(object sender, EventArgs e)

{

if (!string.IsNullOrEmpty(txt\_ogrAtma\_id.Text) && int.TryParse(txt\_ogrAtma\_id.Text, out int ogrAtamaId))

{

dtogratama.DeleteOgrAtama(ogrAtamaId);

MessageBox.Show("Oğretmen ataması " + ogrAtamaId + " silindi", "bilgi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

OgretmenAtamaListesi();

}

else

{

MessageBox.Show("Geçersiz atama ID", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void OgretmenAtamaListesi()

{

dataGridView3.DataSource = dtOgrAtamaJoin.GetOgrAtamaJoin();

}

private void OgretmenMusaitlikListesi()

{

dataGridView1.DataSource=dtOgrMusaitlikJoin.GetOgretmenMusaitlikJoin();

}

private void btn\_kaydet\_Click(object sender, EventArgs e)

{

try

{

SecilenOgrId = Convert.ToInt32(cmbx\_ogrmusait.SelectedValue);

dtogrMusaitlik.Insert(SecilenOgrId, txt\_gun.Text);

MessageBox.Show("Ogretmen müsait işlemi yapıldı", "bilgi", MessageBoxButtons.OK, MessageBoxIcon.Information);

OgretmenMusaitlikListesi();

}

catch (System.Data.SqlClient.SqlException ex)

{

if (ex.Number == 2627)

{

MessageBox.Show("Bu atama zaten yapıldı.", "Hata " + ex.Number, MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else if (ex.Number == 547)

{

MessageBox.Show("Lütfen boş bırakmayın.", "Hata " + ex.Number, MessageBoxButtons.OK, MessageBoxIcon.Error);

}

else

{

MessageBox.Show("Hata ", "Hata " + ex.Number, MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

private void btn\_gunc\_ogr\_belirleme\_Click(object sender, EventArgs e)

{

if (cmbx\_ogrmusait.SelectedItem != null)

{

SecilenOgrId = Convert.ToInt32(cmbx\_ogrmusait.SelectedValue);

}

dtogrMusaitlik.UpdateOgrMusaitlik(SecilenOgrId, txt\_gun.Text, int.Parse(txt\_ogrMusait\_id.Text));

MessageBox.Show("Ogretmen müsaitliği" + txt\_ogrMusait\_id.Text + " güncellendi", "bilgi", MessageBoxButtons.OK, MessageBoxIcon.Information);

OgretmenMusaitlikListesi();

}

private void btn\_sil\_Click(object sender, EventArgs e)

{

if (!string.IsNullOrEmpty(txt\_ogrMusait\_id.Text) && int.TryParse(txt\_ogrMusait\_id.Text, out int ogrMusaitId))

{

dtogrMusaitlik.DeleteOgrMusaitlik(ogrMusaitId);

MessageBox.Show("Oğretmen müsaitliği " + ogrMusaitId + " silindi", "bilgi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

OgretmenMusaitlikListesi();

}

else

{

MessageBox.Show("Geçersiz müsaitlik ID", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void panel1\_Paint(object sender, PaintEventArgs e)

{

}

private void panel2\_Paint(object sender, PaintEventArgs e)

{

}

}

}  
Bu kod bloğu sayesinde artık tablomuzun güncel görüntüsü bu şekilde oluşturulmuştur. Şimdi sırayla butonlara gelelim…

ekran görüntüsü, metin, yazılım, bilgisayar simgesi içeren bir resim

Açıklama otomatik olarak oluşturuldu

## 

### Veriyi Sil:

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

### Veriyi Kayıt Et:

Veriyi kayıt etmek için belirli şartlar ve olaylar vardir. Sırayla inceleyelim..

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

### ***Veriyi Güncelle:***

ekran görüntüsü, metin, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

### ***Cıktı Al:***

Cıktı alma işleminde normal çıktıda tablo gibi bir cıktı alabilmemizi sağlıyor. Aslında tablodaki veriyi cıktı olarak alıyoruz burada…. Gerekli kodları inceleyelim..

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

Bu şekilde excel çıktısı alabiliriz.  
metin, ekran görüntüsü, sayı, numara, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu  
Örnek çıktımız bu şekildedir.

### 

# ***Algoritma Aşamaları Kodlarla***

Şimdi şuana kadar yazılan kodlardaki herşeyi kapataslak anlatmış bulunmaktayız. Şuana kadar kurut işlemlerinin hepsinin nasıl yapılacağını ve yapıldığını gösterdim. Şimdi ders programı otomasyonunu çalıştırmayı anlatacağım.

## ***1. Öğretmenlerin Müsaitliği***

Algoritmada bölümlerin o gün o ders başka ders almaması için müsait olup olmadiğini kontrol edeceğiz algoritma ilk olarak tüm gün müsait olarak düşünecek ve nodelere o şekilde kayıt edecek daha sonrasında ders atadıkça bu kaydı silecek.  
metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

ekran görüntüsü, metin, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

## ***4. Ders Bilgileri***

Şimdi ders bilgilerini kayıt etmek için bir node oluşturacağız..  
metin, ekran görüntüsü, yazılım, ekran, görüntüleme içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım, multimedya yazılımı içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, yazılım içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

## Alogirtma Akış Diyagramı…

* 1. Bölümlerin okul acık olduğu tüm gün müsait olarak kayıt et.
  2. Derslikleri okul acık olduğu tüm gün müsait olarak kayıt et.
  3. Öğretmenlerin okul acık olduğu tüm gün müsait olarak kayıt et.
  4. Öğretmenlerde müsait olmadiği günleri sil.
  5. Derslerin gerekli bilgilerini al.
  6. Aldiğin bilgileri topla ve sırayla oku.
  7. Uygun öğretmen ve bölümü uygun saatlerini bul.
  8. Uygun Derslik bul.
  9. Uygun zamanı bölüm, öğretmen, derslik kayıt et.
  10. Bölüm ve öğretmen, derslik müsaitliğini sil o gün için