public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

//------------------

SqlConnection bag = new SqlConnection(@"Data Source=.\SQLEXPRESS;Initial Catalog=okul;Integrated Security=True");

// SQL Server'a bağlanmak için kullanılan bağlantı nesnesi oluşturulur.

// "Data Source=.\SQLEXPRESS": SQL Server örneğinin adı belirtilir (yerel bilgisayarda SQLEXPRESS).

// "Initial Catalog=okul": Bağlanılacak veritabanının adı "okul" olarak belirtilir.

// "Integrated Security=True": Windows Kimlik Doğrulaması kullanılarak bağlantı yapılacağını belirtir (kullanıcı adı ve parola yerine Windows kimlik bilgileri kullanılır).

//-------------------

private void tabControl1\_SelectedIndexChanged(object sender, EventArgs e)

{

dataGridView1.Columns.Clear(); // DataGridView'deki tüm sütunları temizler.

}

//---------------------------

private void button3\_Click\_1(object sender, EventArgs e)

{

if (!OkulNumarasiDoldurulmus())

{

// Eğer OkulNumarasiDoldurulmus() metodu false dönerse (okul numarası girilmemişse)

MessageBox.Show("Lütfen okul numarasını girin.", "Eksik Bilgi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

// Kullanıcıya okul numarasını girmesi gerektiğini belirten bir mesaj kutusu gösterir.

return; // İşlemi sonlandırır ve kodun geri kalanı çalıştırılmaz.

}

try

{

bag.Open(); // Veritabanı bağlantısını açar.

SqlCommand komut = new SqlCommand("select \* from ogrenci where okulno=@okulno", bag);

// Belirli bir okul numarasına göre veritabanındaki öğrenci bilgilerini seçen bir SQL komutu oluşturur.

komut.Parameters.AddWithValue("@okulno", textBox1.Text); // SQL komutuna okul numarasını parametre olarak ekler.

SqlDataReader oku = komut.ExecuteReader(); // SQL komutunu çalıştırır ve sonucu okuyucuya (reader) atar.

if (oku.HasRows)

{

while (oku.Read()) // Okuyucu verileri okurken

{

// TextBox'lara ve diğer kontrollerdeki değerleri doldurur

textBox2.Text = oku["okulno"].ToString();

textBox3.Text = oku["ad"].ToString();

textBox4.Text = oku["soyad"].ToString();

textBox5.Text = oku["sinif"].ToString();

textBox6.Text = oku["tc\_no"].ToString();

textBox7.Text = oku["cinsiyet"].ToString();

string resimYolu = oku["resim"].ToString();

if (System.IO.File.Exists(resimYolu)) // Resim dosyasının var olup olmadığını kontrol eder

{

pictureBox2.Image = Image.FromFile(resimYolu); // Resim dosyasını PictureBox'a yükler

}

else

{

pictureBox2.Image = null; // Resim dosyası bulunamazsa PictureBox'ı temizler

MessageBox.Show("Resim dosyası bulunamadı."); // Kullanıcıya resim dosyasının bulunamadığını belirten bir mesaj gösterir

}

}

}

else

{

MessageBox.Show("Böyle bir Kayıt Yok!"); // Kayıt bulunamazsa kullanıcıya uyarı mesajı gösterir

textBox1.Clear(); // textBox1'i temizler

textBox2.Clear(); // textBox2'yi temizler

textBox3.Clear(); // textBox3'ü temizler

textBox4.Clear(); // textBox4'ü temizler

textBox5.Clear(); // textBox5'i temizler

textBox6.Clear(); // textBox6'yı temizler

textBox7.Clear(); // textBox7'yi temizler

pictureBox2.Image = null; // pictureBox2'yi temizler

}

}

catch (Exception ex)

{

MessageBox.Show("Hata: " + ex.Message); // Hata olursa hata mesajı gösterir

}

finally

{

bag.Close(); // Veritabanı bağlantısını kapatır

}

}

private bool OkulNumarasiDoldurulmus()

{

// textBox1'in metninin boş olup olmadığını kontrol eder

if (string.IsNullOrWhiteSpace(textBox1.Text)) // textBox1'in metni boşsa veya sadece boşluklardan oluşuyorsa

{

return false; // false döner

}

return true; // textBox1'in metni doluysa, true döner

}

//-------------------------

private void button1\_Click(object sender, EventArgs e)

{

if (!TümAlanlarDoldurulmuş())

{

// Eğer TümAlanlarDoldurulmuş() metodu false dönerse (tüm gerekli alanlar doldurulmamışsa)

MessageBox.Show("Lütfen tüm alanları doldurun.", "Eksik Bilgi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

// Kullanıcıya eksik bilgileri doldurması gerektiğini belirten bir mesaj kutusu gösterir.

return; // İşlemi sonlandırır ve kodun geri kalanı çalıştırılmaz.

}

try

{

bag.Open(); // Veritabanı bağlantısını açar.

SqlCommand komut = new SqlCommand("insert into ogrenci(okulno,ad,soyad,sinif,tc\_no,cinsiyet,resim) values (@p1,@p2,@p3,@p4,@p5,@p6,@p7)", bag);

// Yeni bir SQL INSERT komutu oluşturur ve parametreleri ekler.

komut.Parameters.AddWithValue("@p1", Convert.ToInt32(maskedTextBox1.Text)); // İlk parametre: okul no (int)

komut.Parameters.AddWithValue("@p2", maskedTextBox2.Text); // İkinci parametre: ad (string)

komut.Parameters.AddWithValue("@p3", maskedTextBox3.Text); // Üçüncü parametre: soyad (string)

komut.Parameters.AddWithValue("@p4", maskedTextBox4.Text); // Dördüncü parametre: sınıf (string)

komut.Parameters.AddWithValue("@p5", maskedTextBox5.Text); // Beşinci parametre: TC no (int)

komut.Parameters.AddWithValue("@p6", maskedTextBox6.Text); // Altıncı parametre: cinsiyet (string)

komut.Parameters.AddWithValue("@p7", maskedTextBox7.Text); // Yedinci parametre: resim yolu (string)

komut.ExecuteNonQuery(); // SQL komutunu çalıştırır ve veritabanına kaydı ekler.

MessageBox.Show("Kayıt Eklendi."); // Kayıt eklendi mesajı gösterir.

}

catch (Exception ex)

{

MessageBox.Show("Hata: " + ex.Message); // Hata olursa hata mesajı gösterir.

}

finally

{

bag.Close(); // Veritabanı bağlantısını kapatır.

maskedTextBox1.Clear(); // maskedTextBox1'i temizler.

maskedTextBox2.Clear(); // maskedTextBox2'yi temizler.

maskedTextBox3.Clear(); // maskedTextBox3'ü temizler.

maskedTextBox4.Clear(); // maskedTextBox4'ü temizler.

maskedTextBox5.Clear(); // maskedTextBox5'i temizler.

maskedTextBox6.Clear(); // maskedTextBox6'yı temizler.

maskedTextBox7.Clear(); // maskedTextBox7'yi temizler.

pictureBox1.Image = null; // PictureBox'taki görüntüyü temizler.

}

}

private bool TümAlanlarDoldurulmuş()

{

// Belirtilen TextBox'ların metinlerinin boş olup olmadığını kontrol eder

if (string.IsNullOrWhiteSpace(maskedTextBox1.Text) || // maskedTextBox1'in metni boşsa veya sadece boşluklardan oluşuyorsa

string.IsNullOrWhiteSpace(maskedTextBox2.Text) || // maskedTextBox2'nin metni boşsa veya sadece boşluklardan oluşuyorsa

string.IsNullOrWhiteSpace(maskedTextBox3.Text) || // maskedTextBox3'ün metni boşsa veya sadece boşluklardan oluşuyorsa

string.IsNullOrWhiteSpace(maskedTextBox4.Text) || // maskedTextBox4'ün metni boşsa veya sadece boşluklardan oluşuyorsa

string.IsNullOrWhiteSpace(maskedTextBox5.Text) || // maskedTextBox5'in metni boşsa veya sadece boşluklardan oluşuyorsa

string.IsNullOrWhiteSpace(maskedTextBox6.Text) || // maskedTextBox6'nın metni boşsa veya sadece boşluklardan oluşuyorsa

string.IsNullOrWhiteSpace(maskedTextBox7.Text)) // maskedTextBox7'nin metni boşsa veya sadece boşluklardan oluşuyorsa

{

return false; // Eğer herhangi bir TextBox boşsa veya sadece boşluklardan oluşuyorsa, false döner

}

return true; // Tüm TextBox'lar doluysa, true döner

}

//-----------------------

private void button2\_Click(object sender, EventArgs e)

{

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

// Eğer kullanıcı bir dosya seçer ve "OK" düğmesine tıklarsa

pictureBox1.ImageLocation = openFileDialog1.FileName; // Seçilen dosyanın yolunu PictureBox'ın ImageLocation özelliğine atar ve görüntüyü gösterir.

maskedTextBox7.Text = openFileDialog1.FileName; // Seçilen dosyanın yolunu textBox26'ya yazar.

}

}

//------------------------

private void button5\_Click(object sender, EventArgs e)

{

if (!OkulNumarasiDoldurulmus1())

{

// Eğer OkulNumarasiDoldurulmus1() metodu false dönerse (okul numarası girilmemişse)

MessageBox.Show("Lütfen okul numarasını girin.", "Eksik Bilgi", MessageBoxButtons.OK, MessageBoxIcon.Warning);

// Kullanıcıya okul numarasını girmesi gerektiğini belirten bir mesaj kutusu gösterir.

return; // İşlemi sonlandırır ve kodun geri kalanı çalıştırılmaz.

}

try

{

bag.Open(); // Veritabanı bağlantısını açar.

SqlCommand komut = new SqlCommand("select \* from ogrenci where okulno=@okulno", bag);

// Belirli bir okul numarasına göre veritabanındaki öğrenci bilgilerini seçen bir SQL komutu oluşturur.

komut.Parameters.AddWithValue("@okulno", textBox33.Text); // SQL komutuna okul numarasını parametre olarak ekler.

SqlDataReader oku = komut.ExecuteReader(); // SQL komutunu çalıştırır ve sonucu okuyucuya (reader) atar.

if (oku.HasRows)

{

while (oku.Read()) // Okuyucu verileri okurken

{

// TextBox'lara ve diğer kontrollerdeki değerleri doldurur

textBox32.Text = oku["okulno"].ToString();

textBox31.Text = oku["ad"].ToString();

textBox30.Text = oku["soyad"].ToString();

textBox29.Text = oku["sinif"].ToString();

textBox28.Text = oku["tc\_no"].ToString();

textBox27.Text = oku["cinsiyet"].ToString();

string resimYolu = oku["resim"].ToString();

if (System.IO.File.Exists(resimYolu)) // Resim dosyasının var olup olmadığını kontrol eder

{

pictureBox3.Image = Image.FromFile(resimYolu); // Resim dosyasını PictureBox'a yükler

}

else

{

pictureBox3.Image = null; // Resim dosyası bulunamazsa PictureBox'ı temizler

MessageBox.Show("Resim dosyası bulunamadı."); // Kullanıcıya resim dosyasının bulunamadığını belirten bir mesaj gösterir

}

}

}

else

{

MessageBox.Show("Böyle bir Kayıt Yok!"); // Kayıt bulunamazsa kullanıcıya uyarı mesajı gösterir

textBox33.Clear(); // textBox33'ü temizler

}

}

catch (Exception ex)

{

MessageBox.Show("Hata: " + ex.Message); // Hata olursa hata mesajı gösterir

}

finally

{

bag.Close(); // Veritabanı bağlantısını kapatır

}

}

private bool OkulNumarasiDoldurulmus1()

{

return !string.IsNullOrWhiteSpace(textBox33.Text); // textBox33'ün metni boş değilse veya sadece boşluklardan oluşmuyorsa true döner, aksi takdirde false döner.

}

//-------------------

private void button6\_Click(object sender, EventArgs e)

{

DialogResult cik = MessageBox.Show("Silme işlemine devam etmek istiyor musunuz?", "Uyarı", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

// Kullanıcıya bir silme işlemini onaylaması için mesaj kutusu gösterir ve kullanıcının cevabını DialogResult türünde saklar.

if (cik == DialogResult.Yes)

{

try

{

bag.Open(); // Veritabanı bağlantısını açar.

SqlCommand sil = new SqlCommand("delete FROM ogrenci where okulno=@okulno", bag);

// Belirtilen okul numarasına göre öğrenci kaydını silen bir SQL DELETE komutu oluşturur.

sil.Parameters.AddWithValue("@okulno", Convert.ToInt16(textBox33.Text)); // SQL komutuna okul numarasını parametre olarak ekler.

sil.ExecuteNonQuery(); // SQL komutunu çalıştırır ve öğrenci kaydını siler.

MessageBox.Show("Silme işlemi başarılı"); // Silme işleminin başarılı olduğunu belirten mesaj gösterir.

// TextBox'ları ve PictureBox'ı temizleme

textBox32.Clear(); // textBox32'yi temizler.

textBox31.Clear(); // textBox31'i temizler.

textBox30.Clear(); // textBox30'u temizler.

textBox29.Clear(); // textBox29'u temizler.

textBox28.Clear(); // textBox28'i temizler.

textBox27.Clear(); // textBox27'yi temizler.

textBox33.Clear(); // textBox33'ü temizler.

pictureBox3.Image = null; // pictureBox3'teki görüntüyü temizler.

}

catch (Exception ex)

{

MessageBox.Show("Hata: " + ex.Message); // Hata olursa hata mesajı gösterir.

}

finally

{

bag.Close(); // Veritabanı bağlantısını kapatır.

textBox33.Enabled = true; // textBox33'ü etkin hale getirir.

}

}

else

{

MessageBox.Show("Silme işlemi gerçekleştirilmedi"); // Kullanıcı silme işlemini onaylamazsa, işlem gerçekleştirilmez ve mesaj gösterilir.

}

}

//------------------

private void Listele\_Click(object sender, EventArgs e)

{

bag.Open(); // Veritabanı bağlantısını açar.

string getir = "Select \* From ogrenci"; // Veritabanından tüm öğrenci kayıtlarını seçen SQL sorgusu.

SqlCommand komut = new SqlCommand(getir, bag); // SQL komutunu ve bağlantısını kullanarak yeni bir SqlCommand nesnesi oluşturur.

SqlDataAdapter ad = new SqlDataAdapter(komut); // Verileri almak için SqlCommand nesnesini kullanan SqlDataAdapter nesnesi oluşturur.

DataTable dt = new DataTable(); // Verileri saklamak için yeni bir DataTable nesnesi oluşturur.

ad.Fill(dt); // SqlDataAdapter kullanarak verileri DataTable nesnesine doldurur.

dataGridView1.DataSource = dt; // DataGridView'in veri kaynağını doldurulan DataTable nesnesi olarak ayarlar.

bag.Close(); // Veritabanı bağlantısını kapatır.

}

//--------------------

private void Form1\_Load\_1(object sender, EventArgs e)

{

textBox32.ReadOnly = true; // textBox32 sadece okunabilir hale getirilir (kullanıcılar tarafından düzenlenemez).

textBox31.ReadOnly = true; // textBox31 sadece okunabilir hale getirilir.

textBox30.ReadOnly = true; // textBox30 sadece okunabilir hale getirilir.

textBox29.ReadOnly = true; // textBox29 sadece okunabilir hale getirilir.

textBox28.ReadOnly = true; // textBox28 sadece okunabilir hale getirilir.

textBox27.ReadOnly = true; // textBox27 sadece okunabilir hale getirilir.

textBox2.ReadOnly = true; // textBox2 sadece okunabilir hale getirilir.

textBox3.ReadOnly = true; // textBox3 sadece okunabilir hale getirilir.

textBox4.ReadOnly = true; // textBox4 sadece okunabilir hale getirilir.

textBox5.ReadOnly = true; // textBox5 sadece okunabilir hale getirilir.

textBox6.ReadOnly = true; // textBox6 sadece okunabilir hale getirilir.

textBox7.ReadOnly = true; // textBox7 sadece okunabilir hale getirilir.

maskedTextBox1.Mask = "00000000";

maskedTextBox2.Mask = "LLLLLLLLLLLLLLLLLLLLLLLLLLLLLL";

maskedTextBox3.Mask = "LLLLLLLLLLLLLLLLLLLL";

maskedTextBox4.Mask = "00-L";

maskedTextBox5.Mask = "00000000000";

maskedTextBox6.Mask = "L";

tabControl1.SelectedIndexChanged += new EventHandler(tabControl1\_SelectedIndexChanged); // TabControl'deki seçili sekme değiştiğinde tetiklenecek olay işleyiciyi ekler.

}

}