**PROJE AÇIK KAYNAK KODLARI**

**RestOto**

**YUNUS ORUÇ**

**202106041**

**Amaç:** Bu rapor RestOto Projesinin arayüz tasarımı ile birlikte proje kodlarını göstermeyi amaçlamaktadır. Tasarım ve yazılım sürecinin belgelenmesini sağlamaktadır.

**Sistem Gereksinimleri:** RestOto Projesi windows masaüstü uygulaması olarak tasarlanmıştır. Proje tüm windows işletim sistemlerini desteklemektedir.

**Tasarım:** RestOto Projesi “Visual Studio” sisteminde “C#” yazılım dilini kullanarak tasarlanmıştır. Veri tabanı sisteminde ise “SQL Management Studio” programı kullanılmıştır.

**Tasarım Formları ve Kodlar**

1. **GirisFormu**

**metin, ekran görüntüsü, yazı tipi, logo içeren bir resim

Açıklama otomatik olarak oluşturuldu**

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 RESTOTOo

{

public partial class GirisFormu : Form

{

public GirisFormu()

{

InitializeComponent();

GirisParolaTbox.UseSystemPasswordChar = true;

}

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

{

GirisKullaniciAdiTBox.Select();

}

static string constring = @"Data Source=DESKTOP-8BVLFOK\SQLEXPRESS;Initial Catalog=RestOto;User ID=sa;Password='sa'"; SqlConnection baglan = new SqlConnection(constring);

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

{

string kullaniciAdi = GirisKullaniciAdiTBox.Text;

string parola = GirisParolaTbox.Text;

if (KullaniciDogrula(kullaniciAdi, parola))

{

this.Hide();

AnaForm anaForm = new AnaForm();

anaForm.Show();

}

else

{

MessageBox.Show("Kullanıcı Adı veya Parola Yanlış");

}

}

private bool KullaniciDogrula(string kullaniciAdi, string parola)

{

using (SqlConnection baglan = new SqlConnection(constring))

{

try

{

baglan.Open();

string sorgu = "SELECT COUNT(\*) FROM KullaniciTb WHERE KullaniciAdi = @kullaniciadi AND KullaniciParola = @kullaniciparola";

SqlCommand komut = new SqlCommand(sorgu, baglan);

komut.Parameters.AddWithValue("@kullaniciadi", kullaniciAdi);

komut.Parameters.AddWithValue("@kullaniciparola", parola);

int kullaniciSayisi = (int)komut.ExecuteScalar();

return kullaniciSayisi > 0;

}

catch (Exception ex)

{

MessageBox.Show("Bir Hata Oluştu: " + ex.Message);

return false;

}

}

}

}

}

**2) AnaForm**

metin, çizgi film, ekran görüntüsü, kırpıntı çizim içeren bir resim

Açıklama otomatik olarak oluşturuldu

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 RESTOTOo

{

public partial class AnaForm : Form

{

public AnaForm()

{

InitializeComponent();

}

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

{

GelirForm gelirForm = new GelirForm();

gelirForm.Show();

this.Hide();

}

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

{

GiderForm GiderForm = new GiderForm();

GiderForm.Show();

this.Hide();

}

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

{

UrunForm urunForm = new UrunForm();

urunForm.Show();

this.Hide();

}

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

{

TedarikciForm tedarikciForm = new TedarikciForm();

tedarikciForm.Show();

this.Hide();

}

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

{

PersonelForm PersonelForm = new PersonelForm();

PersonelForm.Show();

this.Hide();

}

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

{

KullanıcıForm kullaniciForm = new KullanıcıForm();

kullaniciForm.Show();

this.Hide();

}

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

{

DialogResult result = MessageBox.Show("Sistemden çıkmak istediğinize emin misiniz?", "Uyarı", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (result == DialogResult.Yes)

{

Application.Exit();

}

}

} }

**3) GelirFormu**

**metin, ekran görüntüsü, yazılım, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturuldu**

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 RESTOTOo

{

public partial class GelirForm : Form

{

public GelirForm()

{

InitializeComponent();

}

static string constring = @"Data Source=DESKTOP-8BVLFOK\SQLEXPRESS;Initial Catalog=RestOto;User ID=sa;Password='sa'";

SqlConnection baglan = new SqlConnection(constring);

public void kayitgetir()

{

string getir = "Select \* From GelirTb";

SqlCommand komut = new SqlCommand(getir, baglan);

SqlDataAdapter ad = new SqlDataAdapter(komut);

DataTable dt = new DataTable();

ad.Fill(dt);

GelirDataGridView.DataSource = dt;

baglan.Close();

}

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

{

this.Hide();

AnaForm anaForm = new AnaForm();

anaForm.Show();

}

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

{

kayitgetir();

GelirIdTBox.Clear();

GelirMiktarTBox.Clear();

GelirAciklamaTBox.Clear();

}

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string kaydet = "INSERT INTO GelirTb (GelirMiktari, GelirTarihi, GelirAciklamasi) " +

"VALUES (@gelirmiktari, @gelirtarihi, @geliraciklamasi)";

SqlCommand komut = new SqlCommand(kaydet, baglan);

komut.Parameters.AddWithValue("@gelirmiktari", GelirMiktarTBox.Text);

komut.Parameters.AddWithValue("@gelirtarihi", GelirdateTimePicker.Value);

komut.Parameters.AddWithValue("@geliraciklamasi", GelirAciklamaTBox.Text);

komut.ExecuteNonQuery();

GelirMiktarTBox.Clear();

GelirAciklamaTBox.Clear();

GelirIdTBox.Clear();

MessageBox.Show("Gelir Kaydedildi");

kayitgetir();

}

}

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

}

}

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string sil = "DELETE FROM GelirTb WHERE GelirId = @gelirid";

SqlCommand komut = new SqlCommand(sil, baglan);

komut.Parameters.AddWithValue("@gelirid", GelirIdTBox.Text);

int etkilenenSatir = komut.ExecuteNonQuery();

if (etkilenenSatir > 0)

{

kayitgetir();

MessageBox.Show("Gelir Silindi");

GelirIdTBox.Clear();

}

else

{

MessageBox.Show("Gelir Bulunamadı");

}

}

}

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

}

}

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string guncelle = "UPDATE GelirTb SET ";

if (!string.IsNullOrEmpty(GelirMiktarTBox.Text))

guncelle += "GelirMiktari = @gelirmiktari, ";

if (!string.IsNullOrEmpty(GelirdateTimePicker.Text))

guncelle += "GelirTarihi = @gelirtarihi, ";

if (!string.IsNullOrEmpty(GelirAciklamaTBox.Text))

guncelle += "GelirAciklamasi = @geliraciklamasi, ";

guncelle = guncelle.TrimEnd(',', ' ');

guncelle += " WHERE GelirId = @gelirid";

SqlCommand komut = new SqlCommand(guncelle, baglan);

komut.Parameters.AddWithValue("@gelirid", GelirIdTBox.Text);

komut.Parameters.AddWithValue("@gelirmiktari", GelirMiktarTBox.Text);

komut.Parameters.AddWithValue("@gelirtarihi", GelirdateTimePicker.Value);

komut.Parameters.AddWithValue("@geliraciklamasi", GelirAciklamaTBox.Text);

int etkilenenSatir = komut.ExecuteNonQuery();

if (etkilenenSatir > 0)

{

MessageBox.Show("Gelir Güncellendi");

kayitgetir();

GelirIdTBox.Clear();

GelirMiktarTBox.Clear();

GelirAciklamaTBox.Clear();

}

else

{

MessageBox.Show("Gelir Bulunamadı");

}

}

}

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

}

}

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string ara = "SELECT \* FROM GelirTb WHERE GelirMiktari LIKE '%' + @aranacakKelime + '%'";

SqlCommand komut = new SqlCommand(ara, baglan);

komut.Parameters.AddWithValue("@aranacakKelime", GelirMiktarTBox.Text);

SqlDataAdapter adapter = new SqlDataAdapter(komut);

DataTable dt = new DataTable();

adapter.Fill(dt);

if (dt.Rows.Count > 0)

{

GelirDataGridView.DataSource = dt;

}

else

{

GelirDataGridView.DataSource = null;

MessageBox.Show("Gelir Bulunamadı");

}

}

}

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

}

}

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

// Gelirlerin toplamını hesapla

string gelirSorgusu = "SELECT SUM(GelirMiktari) FROM GelirTb";

SqlCommand gelirKomut = new SqlCommand(gelirSorgusu, baglan);

decimal toplamGelir = (decimal)gelirKomut.ExecuteScalar();

// Giderlerin toplamını hesapla

string giderSorgusu = "SELECT SUM(GiderMiktari) FROM GiderTb";

SqlCommand giderKomut = new SqlCommand(giderSorgusu, baglan);

decimal toplamGider = (decimal)giderKomut.ExecuteScalar();

// Hesapla ve sonucu metin kutusuna yaz

decimal netGelir = toplamGelir - toplamGider;

GelirHesaplaTBox.Text = netGelir.ToString();

}

}

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

}

}

}

}

**4) GiderFormu**

**metin, ekran görüntüsü, yazılım, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturuldu**

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 RESTOTOo

{

public partial class GiderForm : Form

{

public GiderForm()

{

InitializeComponent();

}

static string constring = @"Data Source=DESKTOP-8BVLFOK\SQLEXPRESS;Initial Catalog=RestOto;User ID=sa;Password='sa'";

SqlConnection baglan = new SqlConnection(constring);

public void kayitgetir()

{

string getir = "Select \* From GiderTb";

SqlCommand komut = new SqlCommand(getir, baglan);

SqlDataAdapter ad = new SqlDataAdapter(komut);

DataTable dt = new DataTable();

ad.Fill(dt);

GiderDataGridView.DataSource = dt;

baglan.Close(); }

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

{

this.Hide();

AnaForm anaForm = new AnaForm();

anaForm.Show();

}

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

{

kayitgetir();

GiderIdTBox.Clear();

GiderMiktarTBox.Clear();

GiderAciklamaTBox.Clear();

}

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string kaydet = "INSERT INTO GiderTb (GiderMiktari, GiderTarihi, GiderAciklamasi) " +

"VALUES (@gidermiktari, @gidertarihi, @gideraciklamasi)";

SqlCommand komut = new SqlCommand(kaydet, baglan);

komut.Parameters.AddWithValue("@gidermiktari", GiderMiktarTBox.Text);

komut.Parameters.AddWithValue("@gidertarihi", GiderdateTimePicker.Value);

komut.Parameters.AddWithValue("@gideraciklamasi", GiderAciklamaTBox.Text);

komut.ExecuteNonQuery();

GiderMiktarTBox.Clear();

GiderIdTBox.Clear();

GiderAciklamaTBox.Clear();

MessageBox.Show("Gider Kaydedildi");

kayitgetir();

}

}

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

}

}

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string sil = "DELETE FROM GiderTb WHERE GiderId = @giderid";

SqlCommand komut = new SqlCommand(sil, baglan);

komut.Parameters.AddWithValue("@giderid", GiderIdTBox.Text);

int etkilenenSatir = komut.ExecuteNonQuery();

if (etkilenenSatir > 0)

{

kayitgetir();

MessageBox.Show("Gider Silindi");

GiderIdTBox.Clear();

}

else

{

MessageBox.Show("Gider Bulunamadı");

}

}

}

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

}

}

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string guncelle = "UPDATE GiderTb SET ";

if (!string.IsNullOrEmpty(GiderMiktarTBox.Text))

guncelle += "GiderMiktari = @gidermiktari, ";

if (!string.IsNullOrEmpty(GiderdateTimePicker.Text))

guncelle += "GiderTarihi = @gidertarihi, ";

if (!string.IsNullOrEmpty(GiderAciklamaTBox.Text))

guncelle += "GiderAciklamasi = @gideraciklamasi, ";

guncelle = guncelle.TrimEnd(',', ' ');

guncelle += " WHERE GiderId = @giderid";

SqlCommand komut = new SqlCommand(guncelle, baglan);

komut.Parameters.AddWithValue("@giderid", GiderIdTBox.Text);

komut.Parameters.AddWithValue("@gidermiktari", GiderMiktarTBox.Text);

komut.Parameters.AddWithValue("@gidertarihi", GiderdateTimePicker.Value);

komut.Parameters.AddWithValue("@gideraciklamasi", GiderAciklamaTBox.Text);

int etkilenenSatir = komut.ExecuteNonQuery();

if (etkilenenSatir > 0)

{

MessageBox.Show("Gider Güncellendi");

kayitgetir();

GiderIdTBox.Clear();

GiderMiktarTBox.Clear();

GiderAciklamaTBox.Clear();

}

else

{

MessageBox.Show("Gider Bulunamadı");

} } }

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message); } }

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string ara = "SELECT \* FROM GiderTb WHERE GiderMiktari LIKE '%' + @aranacakKelime + '%'";

SqlCommand komut = new SqlCommand(ara, baglan);

komut.Parameters.AddWithValue("@aranacakKelime", GiderMiktarTBox.Text);

SqlDataAdapter adapter = new SqlDataAdapter(komut);

DataTable dt = new DataTable();

adapter.Fill(dt);

if (dt.Rows.Count > 0)

{

GiderDataGridView.DataSource = dt;

}

else

{

GiderDataGridView.DataSource = null;

MessageBox.Show("Gider Bulunamadı");

}

}

}

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

}

}

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

// Gelirlerin toplamını hesapla

string gelirSorgusu = "SELECT SUM(GelirMiktari) FROM GelirTb";

SqlCommand gelirKomut = new SqlCommand(gelirSorgusu, baglan);

decimal toplamGelir = (decimal)gelirKomut.ExecuteScalar();

// Giderlerin toplamını hesapla

string giderSorgusu = "SELECT SUM(GiderMiktari) FROM GiderTb";

SqlCommand giderKomut = new SqlCommand(giderSorgusu, baglan);

decimal toplamGider = (decimal)giderKomut.ExecuteScalar();

// Hesapla ve sonucu metin kutusuna yaz

decimal netGelir = toplamGelir - toplamGider;

GiderHesaplaTBox.Text = netGelir.ToString();

}

}

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

}

}

}

}

**5) UrunStokFormu**

**metin, ekran görüntüsü, yazılım, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturuldu**

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 RESTOTOo

{

public partial class UrunForm : Form

{

public UrunForm()

{

InitializeComponent();

}

static string constring = @"Data Source=DESKTOP-8BVLFOK\SQLEXPRESS;Initial Catalog=RestOto;User ID=sa;Password='sa'";

SqlConnection baglan = new SqlConnection(constring);

public void kayitgetir()

{

string getir = "Select \* From UrunTb";

SqlCommand komut = new SqlCommand(getir, baglan);

SqlDataAdapter ad = new SqlDataAdapter(komut);

DataTable dt = new DataTable();

ad.Fill(dt);

UrunDataGridView.DataSource = dt;

baglan.Close(); }

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

{ this.Hide();

AnaForm anaForm = new AnaForm();

anaForm.Show(); }

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

{

kayitgetir();

UrunIdTBox.Clear();

UrunAdiTBox.Clear();

UrunMiktariTBox.Clear();

UrunTedarikciTBox.Clear();

UrunTeslimAlanPersonelTBox.Clear();

UrunAciklamaTBox.Clear();

UrunFiyatiTBox.Clear();

UrunKategorisiTBox.Clear(); }

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

// Önce girdiğiniz UrunAdi'na ait bir kaydın olup olmadığını kontrol edelim

string kontrolSorgusu = "SELECT COUNT(\*) FROM UrunTb WHERE UrunAdi = @urunadi";

SqlCommand kontrolKomut = new SqlCommand(kontrolSorgusu, baglan);

kontrolKomut.Parameters.AddWithValue("@urunadi", UrunAdiTBox.Text);

int kayitSayisi = (int)kontrolKomut.ExecuteScalar();

if (kayitSayisi > 0)

{

// Eğer UrunAdi'na ait bir kayıt bulunuyorsa, miktarı güncelleyelim ve diğer verileri üzerine yazalım

string guncelle = "UPDATE UrunTb SET UrunMiktari = UrunMiktari + @urunmiktari, UrunAciklamasi = @urunaciklama, " +

"UrunAlisFiyati = @urunfiyati, UrunKategorisi = @urunkategorisi, TedarikciId = @tedarikciid, " +

"PersonelId = @personelid WHERE UrunAdi = @urunadi";

SqlCommand guncelleKomut = new SqlCommand(guncelle, baglan);

guncelleKomut.Parameters.AddWithValue("@urunadi", UrunAdiTBox.Text);

guncelleKomut.Parameters.AddWithValue("@urunmiktari", UrunMiktariTBox.Text);

guncelleKomut.Parameters.AddWithValue("@urunaciklama", UrunAciklamaTBox.Text);

guncelleKomut.Parameters.AddWithValue("@urunfiyati", UrunFiyatiTBox.Text);

guncelleKomut.Parameters.AddWithValue("@urunkategorisi", UrunKategorisiTBox.Text);

guncelleKomut.Parameters.AddWithValue("@tedarikciid", UrunTedarikciTBox.Text);

guncelleKomut.Parameters.AddWithValue("@personelid", UrunTeslimAlanPersonelTBox.Text);

guncelleKomut.ExecuteNonQuery();

MessageBox.Show("Ürün Güncellendi");

}

else

{

// Eğer UrunAdi'na ait bir kayıt bulunmuyorsa, yeni bir kayıt ekleyelim

string kaydet = "INSERT INTO UrunTb (UrunAdi, UrunMiktari, UrunAciklamasi, UrunAlisFiyati, UrunKategorisi, TedarikciId, PersonelId) " +

"VALUES (@urunadi, @urunmiktari, @urunaciklama, @urunfiyati, @urunkategorisi, @tedarikciid, @personelid )";

SqlCommand kaydetKomut = new SqlCommand(kaydet, baglan);

kaydetKomut.Parameters.AddWithValue("@urunadi", UrunAdiTBox.Text);

kaydetKomut.Parameters.AddWithValue("@urunmiktari", UrunMiktariTBox.Text);

kaydetKomut.Parameters.AddWithValue("@urunaciklama", UrunAciklamaTBox.Text);

kaydetKomut.Parameters.AddWithValue("@urunfiyati", UrunFiyatiTBox.Text);

kaydetKomut.Parameters.AddWithValue("@urunkategorisi", UrunKategorisiTBox.Text);

kaydetKomut.Parameters.AddWithValue("@tedarikciid", UrunTedarikciTBox.Text);

kaydetKomut.Parameters.AddWithValue("@personelid", UrunTeslimAlanPersonelTBox.Text);

kaydetKomut.ExecuteNonQuery();

MessageBox.Show("Yeni Ürün Kaydedildi");

}

// Alanları temizle ve kayıtları güncelle

UrunAdiTBox.Clear();

UrunMiktariTBox.Clear();

UrunAciklamaTBox.Clear();

UrunFiyatiTBox.Clear();

UrunKategorisiTBox.Clear();

UrunTedarikciTBox.Clear();

UrunTeslimAlanPersonelTBox.Clear();

UrunIdTBox.Clear();

kayitgetir();

}

}

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

}

}

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string sil = "DELETE FROM UrunTb WHERE UrunId = @urunid";

SqlCommand komut = new SqlCommand(sil, baglan);

komut.Parameters.AddWithValue("@urunid", UrunIdTBox.Text);

int etkilenenSatir = komut.ExecuteNonQuery();

if (etkilenenSatir > 0)

{

kayitgetir();

MessageBox.Show("Ürün Silindi");

UrunIdTBox.Clear();

}

else

{

MessageBox.Show("Ürün Bulunamadı");

}

}

}

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

}

}

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string guncelle = "UPDATE UrunTb SET ";

if (!string.IsNullOrEmpty(UrunAdiTBox.Text))

guncelle += "UrunAdi = @urunadi, ";

if (!string.IsNullOrEmpty(UrunMiktariTBox.Text))

guncelle += "UrunMiktari = @urunmiktari, ";

if (!string.IsNullOrEmpty(UrunFiyatiTBox.Text))

guncelle += "UrunAlisFiyati = @urunalisfiyati, ";

if (!string.IsNullOrEmpty(UrunAciklamaTBox.Text))

guncelle += "UrunAciklamasi = @urunaciklamasi, ";

if (!string.IsNullOrEmpty(UrunKategorisiTBox.Text))

guncelle += "UrunKategorisi = @urunkategorisi, ";

guncelle = guncelle.TrimEnd(',', ' ');

guncelle += " WHERE UrunId = @urunid";

SqlCommand komut = new SqlCommand(guncelle, baglan);

komut.Parameters.AddWithValue("@urunid", UrunIdTBox.Text);

komut.Parameters.AddWithValue("@urunadi", UrunAdiTBox.Text);

komut.Parameters.AddWithValue("@urunmiktari", UrunMiktariTBox.Text);

komut.Parameters.AddWithValue("@urunalisfiyati", UrunFiyatiTBox.Text);

komut.Parameters.AddWithValue("@urunaciklamasi", UrunAciklamaTBox.Text);

komut.Parameters.AddWithValue("@urunkategorisi", UrunKategorisiTBox.Text);

int etkilenenSatir = komut.ExecuteNonQuery();

if (etkilenenSatir > 0)

{

MessageBox.Show("Ürün Güncellendi");

kayitgetir();

UrunIdTBox.Clear();

UrunAdiTBox.Clear();

UrunMiktariTBox.Clear();

UrunTedarikciTBox.Clear();

UrunTeslimAlanPersonelTBox.Clear();

UrunAciklamaTBox.Clear();

UrunFiyatiTBox.Clear();

UrunKategorisiTBox.Clear();

}

else

{

MessageBox.Show("Ürün Bulunamadı"); } } }

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

} }

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

{ try {

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string ara = "SELECT \* FROM UrunTb WHERE UrunAdi LIKE '%' + @aranacakKelime + '%'";

SqlCommand komut = new SqlCommand(ara, baglan);

komut.Parameters.AddWithValue("@aranacakKelime", UrunAdiTBox.Text);

SqlDataAdapter adapter = new SqlDataAdapter(komut);

DataTable dt = new DataTable();

adapter.Fill(dt);

if (dt.Rows.Count > 0)

{

UrunDataGridView.DataSource = dt;

}

else

{

UrunDataGridView.DataSource = null;

MessageBox.Show("Ürün Bulunamadı"); } } }

catch (Exception hata)

{ MessageBox.Show("Bir Hata Var! " + hata.Message); } } } }

**6) TedarikciFormu**

metin, ekran görüntüsü, yazılım, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturuldu

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 RESTOTOo

{

public partial class TedarikciForm : Form

{

public TedarikciForm()

{

InitializeComponent();

}

static string constring = @"Data Source=DESKTOP-8BVLFOK\SQLEXPRESS;Initial Catalog=RestOto;User ID=sa;Password='sa'";

SqlConnection baglan = new SqlConnection(constring);

public void kayitgetir()

{

string getir = "Select \* From TedarikciTb";

SqlCommand komut = new SqlCommand(getir, baglan);

SqlDataAdapter ad = new SqlDataAdapter(komut);

DataTable dt = new DataTable();

ad.Fill(dt);

TedarikciDataGridView.DataSource = dt;

baglan.Close();

}

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

{

this.Hide();

AnaForm anaForm = new AnaForm();

anaForm.Show();

}

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

{

kayitgetir();

TedarikciIdTBox.Clear();

TedarikciAdiTBox.Clear();

TedarikciTelTBox.Clear();

TedarikciAdresTBox.Clear();

TedarikciUrunKategorisiTBox.Clear();

}

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string kaydet = "INSERT INTO TedarikciTb (TedarikciAdiSoyadi, TedarikciTelefonu, TedarikciAdresi, UrunKategorisi) " +

"VALUES (@tedarikciadisoyadi, @tedarikcitelefonu, @tedarikciadresi, @urunkategorisi)";

SqlCommand komut = new SqlCommand(kaydet, baglan);

komut.Parameters.AddWithValue("@tedarikciadisoyadi", TedarikciAdiTBox.Text);

komut.Parameters.AddWithValue("@tedarikcitelefonu", TedarikciTelTBox.Text);

komut.Parameters.AddWithValue("@tedarikciadresi", TedarikciAdresTBox.Text);

komut.Parameters.AddWithValue("@urunkategorisi", TedarikciUrunKategorisiTBox.Text);

komut.ExecuteNonQuery();

TedarikciAdiTBox.Clear();

TedarikciTelTBox.Clear();

TedarikciAdresTBox.Clear();

TedarikciUrunKategorisiTBox.Clear();

MessageBox.Show("Tedarikci Kaydedildi");

kayitgetir();

}

}

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

}

}

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string sil = "DELETE FROM TedarikciTb WHERE TedarikciId = @tedarikciid";

SqlCommand komut = new SqlCommand(sil, baglan);

komut.Parameters.AddWithValue("@tedarikciid", TedarikciIdTBox.Text);

int etkilenenSatir = komut.ExecuteNonQuery();

if (etkilenenSatir > 0)

{

kayitgetir();

MessageBox.Show("Tedarikci Silindi");

TedarikciIdTBox.Clear();

}

else

{

MessageBox.Show("Tedarikci Bulunamadı");

}

}

}

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

}

}

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string guncelle = "UPDATE TedarikciTb SET ";

if (!string.IsNullOrEmpty(TedarikciAdiTBox.Text))

guncelle += "TedarikciAdiSoyadi = @tedarikciadisoyadi, ";

if (!string.IsNullOrEmpty(TedarikciTelTBox.Text))

guncelle += "TedarikciTelefonu = @tedarikcitelefonu, ";

if (!string.IsNullOrEmpty(TedarikciAdresTBox.Text))

guncelle += "TedarikciAdresi = @tedarikciadresi, ";

if (!string.IsNullOrEmpty(TedarikciUrunKategorisiTBox.Text))

guncelle += "UrunKategorisi = @urunkategorisi ";

guncelle = guncelle.TrimEnd(',', ' ');

guncelle += " WHERE TedarikciId = @tedarikciid";

SqlCommand komut = new SqlCommand(guncelle, baglan);

komut.Parameters.AddWithValue("@tedarikciid", TedarikciIdTBox.Text);

komut.Parameters.AddWithValue("@tedarikciadisoyadi", TedarikciAdiTBox.Text);

komut.Parameters.AddWithValue("@tedarikcitelefonu", TedarikciTelTBox.Text);

komut.Parameters.AddWithValue("@tedarikciadresi", TedarikciAdresTBox.Text);

komut.Parameters.AddWithValue("@urunkategorisi", TedarikciUrunKategorisiTBox.Text);

int etkilenenSatir = komut.ExecuteNonQuery();

if (etkilenenSatir > 0)

{

MessageBox.Show("Tedarikçi Güncellendi");

kayitgetir();

TedarikciIdTBox.Clear();

TedarikciAdiTBox.Clear();

TedarikciTelTBox.Clear();

TedarikciAdresTBox.Clear();

TedarikciUrunKategorisiTBox.Clear();

}

else

{

MessageBox.Show("Tedarikçi Bulunamadı");

}

}

}

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

}

}

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string ara = "SELECT \* FROM TedarikciTb WHERE TedarikciAdiSoyadi LIKE '%' + @aranacakKelime + '%'";

SqlCommand komut = new SqlCommand(ara, baglan);

komut.Parameters.AddWithValue("@aranacakKelime", TedarikciAdiTBox.Text);

SqlDataAdapter adapter = new SqlDataAdapter(komut);

DataTable dt = new DataTable();

adapter.Fill(dt);

if (dt.Rows.Count > 0)

{

TedarikciDataGridView.DataSource = dt;

}

else

{

TedarikciDataGridView.DataSource = null;

MessageBox.Show("Personel Bulunamadı");

} } }

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

} } } }

**7) TedarikciFormu**

metin, ekran görüntüsü, yazılım, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturuldu

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 RESTOTOo

{

public partial class PersonelForm : Form

{

public PersonelForm()

{

InitializeComponent();

}

static string constring = @"Data Source=DESKTOP-8BVLFOK\SQLEXPRESS;Initial Catalog=RestOto;User ID=sa;Password='sa'";

SqlConnection baglan = new SqlConnection(constring);

public void kayitgetir()

{

string getir = "Select \* From PersonelTb";

SqlCommand komut = new SqlCommand(getir, baglan);

SqlDataAdapter ad = new SqlDataAdapter(komut);

DataTable dt = new DataTable();

ad.Fill(dt);

PersonelDataGridView.DataSource = dt;

baglan.Close();

}

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

{

this.Hide();

AnaForm anaForm = new AnaForm();

anaForm.Show();

}

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

{

kayitgetir();

PersonelIdTBox.Clear();

PersonelAdiTBox.Clear();

PersonelTelNoTBox.Clear();

PersonelMaasiTBox.Clear();

PersonelPozisyonuTBox.Clear();

}

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string kaydet = "INSERT INTO PersonelTb (PersonelAdiSoyadi, PersonelTeli, PersonelMaasi, PersonelPosziyonu, PersonelGirisTarihi) " +

"VALUES (@personeladisoyadi, @personeltei, @personelmaasi, @personelpozisyonu, @personelgiristarihi)";

SqlCommand komut = new SqlCommand(kaydet, baglan);

komut.Parameters.AddWithValue("@personeladisoyadi", PersonelAdiTBox.Text);

komut.Parameters.AddWithValue("@personeltei", PersonelTelNoTBox.Text);

komut.Parameters.AddWithValue("@personelmaasi", PersonelMaasiTBox.Text);

komut.Parameters.AddWithValue("@personelpozisyonu", PersonelPozisyonuTBox.Text);

komut.Parameters.Add("@personelgiristarihi", PersoneldateTimePicker.Value);

komut.ExecuteNonQuery();

PersonelAdiTBox.Clear();

PersonelTelNoTBox.Clear();

PersonelMaasiTBox.Clear();

PersonelPozisyonuTBox.Clear();

MessageBox.Show("Personel Kaydedildi");

kayitgetir();

}

}

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

}

}

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string sil = "DELETE FROM PersonelTb WHERE PersonelId = @personelid";

SqlCommand komut = new SqlCommand(sil, baglan);

komut.Parameters.AddWithValue("@personelid", PersonelIdTBox.Text);

int etkilenenSatir = komut.ExecuteNonQuery();

if (etkilenenSatir > 0)

{

kayitgetir();

MessageBox.Show("Personel Silindi");

PersonelIdTBox.Clear();

}

else

{

MessageBox.Show("Personel Bulunamadı");

}

}

}

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

}

}

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

{

try

{

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string guncelle = "UPDATE PersonelTb SET ";

if (!string.IsNullOrEmpty(PersonelAdiTBox.Text))

guncelle += "PersonelAdiSoyadi = @personeladisoyadi, ";

if (!string.IsNullOrEmpty(PersonelTelNoTBox.Text))

guncelle += "PersonelTeli = @personelteli, ";

if (!string.IsNullOrEmpty(PersonelMaasiTBox.Text))

guncelle += "PersonelMaasi = @personelmaasi, ";

if (!string.IsNullOrEmpty(PersonelPozisyonuTBox.Text))

guncelle += "PersonelPosziyonu = @personelpozisyonu, ";

if (!string.IsNullOrEmpty(PersoneldateTimePicker.Text))

guncelle += "PersonelGirisTarihi = @personelgiristarihi, ";

guncelle = guncelle.TrimEnd(',', ' ');

guncelle += " WHERE PersonelId = @personelid";

SqlCommand komut = new SqlCommand(guncelle, baglan);

komut.Parameters.AddWithValue("@personelid", PersonelIdTBox.Text);

komut.Parameters.AddWithValue("@personeladisoyadi", PersonelAdiTBox.Text);

komut.Parameters.AddWithValue("@personelteli", PersonelTelNoTBox.Text);

komut.Parameters.AddWithValue("@personelmaasi", PersonelMaasiTBox.Text);

komut.Parameters.AddWithValue("@personelpozisyonu", PersonelPozisyonuTBox.Text);

komut.Parameters.AddWithValue("@personelgiristarihi", PersoneldateTimePicker.Value);

int etkilenenSatir = komut.ExecuteNonQuery();

if (etkilenenSatir > 0)

{

MessageBox.Show("Personel Güncellendi");

kayitgetir();

PersonelIdTBox.Clear();

PersonelAdiTBox.Clear();

PersonelTelNoTBox.Clear();

PersonelMaasiTBox.Clear();

PersonelPozisyonuTBox.Clear();

}

else

{

MessageBox.Show("Personel Bulunamadı");

}

} }

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message);

} }

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

{

try {

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string ara = "SELECT \* FROM PersonelTb WHERE PersonelAdiSoyadi LIKE '%' + @aranacakKelime + '%'";

SqlCommand komut = new SqlCommand(ara, baglan);

komut.Parameters.AddWithValue("@aranacakKelime", PersonelAdiTBox.Text);

SqlDataAdapter adapter = new SqlDataAdapter(komut);

DataTable dt = new DataTable();

adapter.Fill(dt);

if (dt.Rows.Count > 0)

{

PersonelDataGridView.DataSource = dt;

}

else

{

PersonelDataGridView.DataSource = null;

MessageBox.Show("Personel Bulunamadı");

} } }

catch (Exception hata)

{

MessageBox.Show("Bir Hata Var! " + hata.Message); } } } }

**8) KullanıcıFormu**

metin, ekran görüntüsü, yazılım, web sayfası içeren bir resim

Açıklama otomatik olarak oluşturuldu

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 RESTOTOo

{

public partial class KullanıcıForm : Form

{

public KullanıcıForm()

{

InitializeComponent();

}

static string constring = @"Data Source=DESKTOP-8BVLFOK\SQLEXPRESS;Initial Catalog=RestOto;User ID=sa;Password='sa'";

SqlConnection baglan = new SqlConnection(constring);

public void kayitgetir()

{

string getir = "Select \* From KullaniciTb";

SqlCommand komut = new SqlCommand(getir, baglan);

SqlDataAdapter ad = new SqlDataAdapter(komut);

DataTable dt = new DataTable();

ad.Fill(dt);

KullaniciDataGridView.DataSource = dt;

baglan.Close();

}

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

{

this.Hide();

AnaForm anaForm = new AnaForm();

anaForm.Show(); }

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

{

kayitgetir(); }

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

{ try {

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string kaydet = "INSERT INTO KullaniciTb (KullaniciAdi, KullaniciParola) " +

"VALUES (@kullaniciadi, @kullaniciparola)";

SqlCommand komut = new SqlCommand(kaydet, baglan);

komut.Parameters.AddWithValue("@kullaniciadi", KullaniciKullaniciAdiTBox.Text);

komut.Parameters.AddWithValue("@kullaniciparola", KullaniciParolaTBox.Text);

komut.ExecuteNonQuery();

KullaniciKullaniciAdiTBox.Clear();

KullaniciParolaTBox.Clear();

MessageBox.Show("Kullanıcı Kaydedildi");

kayitgetir(); } }

catch (Exception hata) {

MessageBox.Show("Bir Hata Var! " + hata.Message);

} }

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

{ try {

if (baglan.State == ConnectionState.Closed)

{

baglan.Open();

string sil = "DELETE FROM KullaniciTb WHERE KullaniciAdi = @kullaniciadi";

SqlCommand komut = new SqlCommand(sil, baglan);

komut.Parameters.AddWithValue("@kullaniciadi", KullaniciKullaniciAdiTBox.Text);

int etkilenenSatir = komut.ExecuteNonQuery();

if (etkilenenSatir > 0)

{

kayitgetir();

MessageBox.Show("Ürün Silindi");

KullaniciKullaniciAdiTBox.Clear(); }

else {

MessageBox.Show("Ürün Bulunamadı"); } } }

catch (Exception hata) {

MessageBox.Show("Bir Hata Var! " + hata.Message); } } } }