

Final Uygulama Projesi Raporu

Yıl / Dönem:

2023-2024 / Güz Yarıyılı

Ders Adı / Ders Kodu:

GÖRSEL PROGRAMLAMA I (BBP201)

Proje Değerlendirme Baremi / Genel Not Baremine Etkisi(%):

Final Uygulaması

Dersin Öğretim Elemanı:

Öğr. Gör. ADNAN KÜRŞAT TEKE

Bölüm/Program:

İGMYO / BİLGİSAYAR PROGRAMCILIĞI (N.Ö.) – A Şubesi

Proje Grubu Adı / Konu:

GELİR GİDER TAKİP ve MUHASEBE UYGULAMASI

Proje Grubu Takım Üyeleri:

220111549-BURAK İŞLER

ÖDEV TANITIM FORMU

YAZAR ADI SOYADI : Burak İŞLER

ÖDEVİN DİLİ : Türkçe

ÖDEVİN ADI : Gelir Gider Takip ve Muhasebe Uygulaması

BÖLÜM : Bilgisayar Teknolojileri

PROGRAM : Bilgisayar Programcılığı

ÖDEVİN TÜRÜ : Final

ÖDEVİN TES. TARİHİ : 29.12.2023

SAYFA SAYISI : 20

ÖDEV DANIŞMANI : Öğr. Gör. Adnan Kürşat TEKE

KABUL VE ONAY SAYFASI

220111549 numaralı Burak İŞLER'in Gelir Gider Takip ve Muhasebe Uygulaması adlı çalışması, benim tarafımdan Final ödevi olarak kabul edilmiştir.

Öğretim Görevlisi
Adnan Kürşat TEKE

ÖZET

Bu projenin amacı, işletmenin gelir gider takibini etkileşimli bir arayüz üzerinden yönetmesini sağlamaktır. Gider Tipi, Gelir Gider Kaydetme, Güncelleme, Silme, Filtreleme ve Listeleme işlemlerini gerçekleştirebilirler.

İÇİNDEKİLER

İçindekiler

ÖDEV TANITIM FORMU	II
KABUL VE ONAY SAYFASI	III
ÖZET	IV
İÇİNDEKİLER	V
ÖNSÖZ	VI
SİSTEM NE İŞE YARIYOR?	1
HEDEF KİTLE	2
ZAMAN ÇİZELGESİ.....	3
PROJE EKİBİ VE YÖNETİMİ	3
KOD ADIMLARI.....	6
KAYNAKÇA.....	9

ÖNSÖZ

Günümüzde bireyler ve işletmeler, finansal durumlarını etkin bir şekilde yönetmek ve geleceklerini planlamak amacıyla gelir ve giderlerini takip etmek zorundadırlar. Bu ihtiyaçtan yola çıkarak geliştirilen "Gelir-Gider ve Muhasebe Takibi" uygulaması, kullanıcılara finansal verilerini düzenli bir şekilde kaydetme, analiz etme ve raporlama imkanı sunmaktadır. Bu rapor, geliştirilen gelir-gider takibi uygulamasının tasarım sürecinden, kullanılan teknolojilere kadar detaylı bir inceleme sunmaktadır. Uygulama, kullanıcı dostu arayüzü, etkili veri yönetimi ve raporlama özellikleri ile finansal takip süreçlerini optimize etmeyi amaçlamaktadır. İstanbul Gelişim Üniversitesi'nde öğretim görevlisi olarak bizlerle bilgi ve birikimini paylaştığı için değerli hocam Adnan Kürşat TEKE'ye teşekkürlerimi sunuyorum.

SİSTEM NE İŞE YARIYOR?

Gelir-Gider Takip ve Muhasebe Sistemi, bireylerin ve işletmelerin finansal durumlarını etkin bir şekilde yönetmelerine yardımcı olan basit ve kullanıcı dostu bir uygulamadır. Bu projenin temel amacı, gelir ve giderleri düzenli bir şekilde kaydetmek, takip etmek ve analiz etmek suretiyle finansal karar alma süreçlerini desteklemektir. Bu projemizde Gider Tipi ve Gelir Gider kaydetme, güncelleme ve silme işlemleri yapılabilir. Ayrıca kayıtlar gelir gider tipine göre listelenebilir ve filtrelenebilir. Olabildiğince basit ve kullanımı kolay hale getirilecek şekilde tasarlanmıştır.

HEDEF KİTLE

Gelir-Gider Takip Uygulamamız, bireysel kullanıcılara kendi finansal durumlarını etkin bir şekilde yönetme ve planlama imkanı sunar. Bu uygulamayı kullanmayı tercih eden bireyler şunlar için idealdir:

1. Bireysel Kullanıcılar
2. Serbest Meslek Sahipleri ve Küçük İşletmeler

ZAMAN ÇİZELGESİ

Gün 1: Proje Planlaması ve Tasarım

Proje gereksinimlerini belirleme ve analiz etme.

Veritabanı yapısını tasarlama.

Kullanıcı arayüzü tasarımına başlama.

Gün 2: Veritabanı Entegrasyonu ve Temel Fonksiyonlar

Entity Framework kullanarak veritabanı bağlantısını kurma.

Gelir ve gider kaydı ekleme, güncelleme ve silme fonksiyonlarını oluşturma.

Temel veri girişi ve çıkışı test etme.

Gün 3: Kategori ve Tip Yönetimi

Gider tipleri ve kategorileri ekleyip düzenleme fonksiyonlarını geliştirme.

Kullanıcıya özelleştirilebilir kategoriler ve tipler ekleme.

Veritabanı üzerinde kategori ve tip yönetimini test etme.

Gün 4: Raporlama ve Analiz

Finansal raporlar oluşturma işlevselliğini ekleyerek, kullanıcıya görsel raporlama sağlama.

Gelir ve gider kayıtlarını filtreleme ve listeletme fonksiyonlarını geliştirme.

Kullanıcı arayüzündeki raporlama öğelerini tasarlayarak entegre etme.

Gün 5: Güvenlik ve Hata Kontrolü

Kullanıcı girişi için basit bir oturum yönetimi ekleyerek güvenliği artırma.

Hata kontrolleri ve kullanıcı dostu hata mesajları ekleyerek uygulamanın sağlamlığını kontrol etme.

Gün 6: İyileştirme ve Dokümantasyon

Uygulamanın performansını iyileştirme ve gereksiz kodları temizleme.

Kullanıcı dostu dokümantasyon ekleyerek, uygulamanın nasıl kullanılacağını anlatma.

Son testleri yaparak uygulamanın kullanıma hazır olduğunu onaylama.

PROJE EKİBİ VE YÖNETİMİ

1. Süleyman Demir:

- **Rol:** Proje Lideri ve Ana Geliştirici
- **Görevler:**
 - Projenin genel yönetiminden sorumlu olacak.
 - Diğer ekip üyeleriyle koordinasyon sağlayacak.
 - Temel C# kodlama ve form tasarımı üzerinde çalışacak.

2. Batuhan Sevin:

- **Rol:** Ana Geliştirici
- **Görevler:**
 - Projenin genel yönetiminden sorumlu olacak.
 - Diğer ekip üyeleriyle koordinasyon sağlayacak.
 - Temel C# kodlama ve form tasarımı üzerinde çalışacak.

3. Burak İşler:

- **Rol:** Ana Geliştirici
- **Görevler:**
 - Projenin genel yönetiminden sorumlu olacak.
 - Diğer ekip üyeleriyle koordinasyon sağlayacak.
 - Temel C# kodlama ve form tasarımı üzerinde çalışacak.

KOD ADIMLARI

Form1 Kodları:

```
using System;
using System.Data;
using System.Data.OleDb;
using System.Windows.Forms;
using System.IO;

namespace Muhasebe
{
    5 bayyuru
    public partial class Form1 : Form
    {
        OleDbConnection baglanti;
        OleDbDataAdapter giderveri, gelirveri;
        OleDbCommand komut;
        DataSet verikumesi;
        DataTable gidertablo, gelirtablo;

        double kdvdahiltutar, kdvd, kdvharcitutar;

        double birimfiyat, adet, toplam;
        2 bayyuru
        public Form1()
        {
            InitializeComponent();
        }

        1 bayyuru
        private void Form1_Load(object sender, EventArgs e)
        {
            this.firmalarTableAdapter.Fill(this.databaseDataSet.firmalar);
            baglanti = new OleDbConnection("Provider=Microsoft.Jet.Oledb.4.0; Data Source=database.mdb");
            VeriCek();
        }

        1 bayyuru
        private void firmakaydet_Click(object sender, EventArgs e)
        {
            baglanti.Open();
            OleDbCommand komut = new OleDbCommand("INSERT INTO firmalar (adi, adres, telefon)" +
            " VALUES ('" + eklefirmaadi.Text.Trim() + "', '" + eklefirmaadres.Text.Trim() + "', '" + eklefirmatelefon.Text.Trim() + "')", baglanti);
            komut.ExecuteNonQuery();
            komut.Dispose();
            baglanti.Close();
            this.firmalarTableAdapter.Fill(this.databaseDataSet.firmalar);
        }

        5 bayyuru
        private void VeriCek()
        {
            giderveri = new OleDbDataAdapter("SELECT * FROM gider", baglanti);
            gelirveri = new OleDbDataAdapter("SELECT * FROM gelir", baglanti);
            gidertablo = new DataTable();
            gelirtablo = new DataTable();
            baglanti.Open();
            giderveri.Fill(gidertablo);
            gelirveri.Fill(gelirtablo);
            baglanti.Close();
            dataGridView1.DataSource = gidertablo;
            dataGridView2.DataSource = gelirtablo;
        }
    }
}
```

Kodumuzun ilk resminde sırasıyla veritabanı bağlantısı, veri adaptörleri, SQL komutu, veri seti, gider ve gelir tabloları, KDV hesaplamak için kullanılan değişkenler, ürün birim fiyatı, adeti ve toplam ücret, form yüklendiğinde yapılacak işlemler, veritabanı bağlantısı oluşturma, veri çekme, firma kaydı ekleme butonu, firma ekleme SQL komutu, firma listesini güncelle, verileri çekme metodu, gider ve gelir verilerini çek, DataTable'ları oluştur, bağlantıyı aç, verileri DataTable'lara doldur, bağlantıyı kapat, DataGridView'lara DataTable'ları ata işlevleri eklenmiştir.

```

1 bagvuru
private void kdvharic_CheckedChanged(object sender, EventArgs e)
{
    KdvHaricHesap();
}

1 bagvuru
private void kdvdahil_CheckedChanged(object sender, EventArgs e)
{
    KdvDahilHesap();
}

1 bagvuru
private void gelirgir_Click(object sender, EventArgs e)
{
    baglanti.Open();
    OleDbCommand komut = new OleDbCommand("INSERT INTO gelir (nerden, neden, kdvdahiltutar, kdvharictutar, tarih, kdv) VALUES " +
        "(" + firmaadi.Text.Trim() + "," + aciklama.Text.Trim() + "," + kdvdahiltutar + "," + kdvharictutar + "," + DateTime.Today.ToString() + "," + kdv + ")", baglanti);
    komut.ExecuteNonQuery();
    komut.Dispose();
    baglanti.Close();
    VeriCek();
}

1 bagvuru
private void button2_Click(object sender, EventArgs e)
{
    Temizle();
}

2 bagvuru
private void Temizle()
{
    listelefirmaadi.Text = "";
    buaykibakiye.Text = "";
    toplambakiye.Text = "";
    VeriCek();
}

1 bagvuru
private void firmasil_Click(object sender, EventArgs e)
{
    baglanti.Open();
    string komut = "DELETE FROM firmalar WHERE adi='" + listelefirmaadi.Text + "'";
    OleDbCommand My_Command = new OleDbCommand(komut, baglanti);
    My_Command.ExecuteNonQuery();
    baglanti.Close();
    this.firmalarTableAdapter.Fill(this.databaseDataSet.firmalar);
    Temizle();
}

1 bagvuru
private void dataGridView1_CellMouseClick(object sender, DataGridViewCellEventArgs e)
{
    if (e.RowIndex >= 0)
    {
        DataGridViewRow row = dataGridView1.Rows[e.RowIndex];
        seciliid.Text = "ID : " + row.Cells[0].Value.ToString();
    }
}

```

Kodumuzun ikinci resminde sırasıyla KDV hesaplama yöntemini seçme radio butonları, gelir girişi butonu, gelir ekleme SQL komutu, verileri çek, temizle butonu, temizleme metodu, firma silme metodu, seçilen firmayı silme SQL komutu, firma listesini güncelle işlevleri eklenmiştir.

```

private void button3_Click(object sender, EventArgs e)
{
    baglanti.Open();
    string komut = "UPDATE gider set durum='Ödendi' WHERE id='" + seciliid.Text + "'";
    OleDbCommand odendiyap = new OleDbCommand(komut, baglanti);
    odendiyap.ExecuteNonQuery();
    baglanti.Close();
    VeriCek();
}

1.bapuru
private void toolStripButton3_Click(object sender, EventArgs e)
{
    Form2 frm = new Form2();
    frm.Show();
}

1.bapuru
private void toolStripButton2_Click(object sender, EventArgs e)
{
    try
    {
        if (File.Exists(Application.StartupPath + "\\database.mdb"))
        {
            saveFileDialog1.DefaultExt = "mdb";
            saveFileDialog1.Filter = "MDB Dosyaları (*.mdb)|*.mdb|Tüm Dosyalar(*.*)|*.*";
            saveFileDialog1.ShowDialog();

            if (saveFileDialog1.FileName != null)
            {
                File.Copy(Application.StartupPath + "\\database.mdb", saveFileDialog1.FileName);
            }
        }
    }
    catch (Exception Ex)
    {
        MessageBox.Show("Hata Oluştı : " + Ex.ToString());
    }
}

1.bapuru
private void toolStripButton1_Click(object sender, EventArgs e)
{
    try
    {
        MessageBox.Show("Lütfen Programın İçinden Yedeklediğiniz Bir Veritabanı (mdb) Dosyasını Seçiniz", "Uyarı", MessageBoxButtons.OKCancel, MessageBoxIcon.Warning);
        openFileDialog1.DefaultExt = "mdb";
        openFileDialog1.Filter = "MDB Dosyaları (*.mdb)|*.mdb|Tüm Dosyalar(*.*)|*.*";
        openFileDialog1.Title = "Veritabanı Seç";
        openFileDialog1.ShowDialog();
        if (openFileDialog1.FileName != null)
        {
            File.Delete(Application.StartupPath + "\\database.mdb");
            File.Copy(openFileDialog1.FileName, Application.StartupPath + "\\database.mdb");
        }
    }
    catch (Exception Ex)
    {
        MessageBox.Show("Hata Oluştı : " + Ex.ToString());
    }
}

```

Kodumuzun üçüncü resminde sırasıyla gider ödendi işaretleme butonu, seçilen giderin durumunu 'Ödendi' olarak güncelleme SQL komutu, verileri çekme, yeni formu açma butonu, veritabanı yedekleme butonu, dosya varsa yedekleme işlemi ve veritabanı geri yükleme butonu işlevleri eklenmiştir.

```

1 başvuru
private void listelefirmaadi_SelectedIndexChanged(object sender, EventArgs e)
{
    ///AY BAKİYE
    baglanti.Open();
    string sql = "SELECT * FROM gider Where tarih BETWEEN @tar1 and @tar2";
    DataTable dt = new DataTable();
    OleDbDataAdapter adp = new OleDbDataAdapter(sql, baglanti);
    adp.SelectCommand.Parameters.AddWithValue("@tar1", DateTime.Today.ToShortDateString());
    adp.SelectCommand.Parameters.AddWithValue("@tar2", DateTime.Today.AddDays(-30).ToShortDateString());
    adp.SelectCommand.Parameters.AddWithValue("@isim", listelefirmaadi.Text);
    adp.Fill(dt);

    string sql2 = "SELECT * FROM gelir WHERE nerden = '" + listelefirmaadi.Text + "'";
    DataTable dt2 = new DataTable();
    OleDbDataAdapter adp2 = new OleDbDataAdapter(sql2, baglanti);
    adp2.Fill(dt2);

    string sql3 = "SELECT * FROM gider WHERE nerden = '" + listelefirmaadi.Text + "'";
    DataTable dt3 = new DataTable();
    OleDbDataAdapter adp3 = new OleDbDataAdapter(sql3, baglanti);
    adp3.Fill(dt3);

    dataGridView1.DataSource = dt;
    //dataGridView1.DataSource = dt;

    int toplam = 0;
    for (int i = 0; i < dataGridView1.Rows.Count; ++i)
    {
        toplam += Convert.ToInt32(dataGridView1.Rows[i].Cells[8].Value);
    }

    buaykibakiye.Text = toplam.ToString() + " TL";
    dataGridView2.DataSource = dt2;

    ///TOPLAM BAKİYE
    string islem2 = "select sum(kdvdahiltutar) from gider WHERE nerden = '" + listelefirmaadi.Text + "'";
    OleDbCommand komut2 = new OleDbCommand(islem2, baglanti);
    toplambakiye.Text = komut2.ExecuteScalar().ToString() + " TL";
    komut2.ExecuteNonQuery(); // Komutu çalıştırıyoruz
    baglanti.Close(); // Bağlantıyı mutlaka kapatıyoruz
}

1 başvuru
private void button1_Click(object sender, EventArgs e)
{
}

1 başvuru
private void birimfiyat_TextChanged(object sender, EventArgs e)
{
    if (miktar.Text != "" && adetfiyat.Text != "")
    {
        birimfiyat = Double.Parse(adetfiyat.Text);
        adet = Double.Parse(miktar.Text);

        toplam = birimfiyat * adet;

        if (kdvdahil.Checked == true)
        {
            KdvDahilHesap();
        }
    }
}

```

Kodumuzun dördüncü resminde sırasıyla firma listesinden firma seçme, ay bazlı bakiye hesaplama, toplam bakiye hesaplama ve birim fiyat girişi değiştiğinde işlevleri eklenmiştir.

```

2 başvuru
private void KdvHaricHesap()
{
    if (kdvorani.Text == "%1")
    {
        kdvtutari.Text = "KDV Tutarı : " + (toplaml * 1 / 100).ToString("#,##0.00") + " TL";
        kdvdahiltutar = toplaml + toplaml * 1 / 100;
        kdvhariictutar = toplaml - toplaml * 1 / 100;
        kdvhariictutar += toplaml * 1 / 100;
        kdvd = toplaml * 18 / 100;
    }
    else if (kdvorani.Text == "%8")
    {
        kdvtutari.Text = "KDV Tutarı : " + (toplaml * 8 / 100).ToString("#,##0.00") + " TL";
        kdvdahiltutar = toplaml + toplaml * 8 / 100;
        kdvhariictutar = toplaml - toplaml * 8 / 100;
        kdvhariictutar += toplaml * 8 / 100;
        kdvd = toplaml * 18 / 100;
    }
    else if (kdvorani.Text == "%18")
    {
        kdvtutari.Text = "KDV Tutarı : " + (toplaml * 18 / 100).ToString("#,##0.00");
        kdvdahiltutar = toplaml + toplaml * 18 / 100;
        kdvhariictutar = toplaml - toplaml * 18 / 100;
        kdvhariictutar += toplaml * 18 / 100;
        kdvd = toplaml * 18 / 100;
    }

    toplamtutar.Text = "Toplam Tutar : " + kdvhariictutar.ToString("#,##0.00") + " TL";
}

2 başvuru
private void KdvDahilHesap()
{
    if (kdvorani.Text == "%1")
    {
        kdvtutari.Text = "KDV Tutarı : " + (toplaml * 1 / 100).ToString("#,##0.00") + " TL";
        kdvdahiltutar = toplaml + toplaml * 1 / 100;
        kdvhariictutar = toplaml - (toplaml * 1 / 100);
        kdvhariictutar += toplaml * 1 / 100;
        kdvd = toplaml * 1 / 100;
    }
    else if (kdvorani.Text == "%8")
    {
        kdvtutari.Text = "KDV Tutarı : " + (toplaml * 8 / 100).ToString("#,##0.00") + " TL";
        kdvdahiltutar = toplaml + toplaml * 8 / 100;
        kdvhariictutar = toplaml - (toplaml * 8 / 100);
        kdvhariictutar += toplaml * 8 / 100;
        kdvd = toplaml * 8 / 100;
    }
    else if (kdvorani.Text == "%18")
    {
        kdvtutari.Text = "KDV Tutarı : " + (toplaml * 18 / 100).ToString("#,##0.00");
        kdvdahiltutar = toplaml + toplaml * 18 / 100;
        kdvhariictutar = toplaml - (toplaml * 18 / 100);
        kdvhariictutar += toplaml * 18 / 100;
        kdvd = toplaml * 18 / 100;
    }
}

```

Kodumuzun beşinci resminde “private void KdvHaricHesap() metodunda “kdvorani” TextBox’ındaki KDV oranına bağlı olarak KDV hesaplamasını gerçekleştirir ve ilgili TextBox’lara sonuçları yazdırır.

private void KdvDahilHesap() metodunda ise “kdvorani” TextBox'ındaki KDV oranına bağılı olarak KDV hesaplamasını gerçekleştirir ve ilgili TextBox'lara sonuçları yazdırır.

Her iki metot da aynı KDV hesaplama mantığına sahiptir, farkları ise KDV'nin nasıl hesaplandığıdır (KdvHaricHesap'ta toplamdan, KdvDahilHesap'ta ise toplama eklenen bir şekilde). Bu metotlar, KDV hesaplamasını ayrı bir fonksiyona taşıyarak kod tekrarını önlemek ve daha okunabilir bir yapı sağlamak için tasarlanmıştır.

KAYNAKÇA

- <https://www.youtube.com/watch?v=VS-Mk6SJwFU>
- <https://mustafabukulmez.com/2021/04/06/c-gelir-gider-takip-uygulamasi-video/>
- <https://youtu.be/0P-i8-3jn-M>
- <https://forum.donanimhaber.com/c-arkadaslar-merhaba-ayilik-ve-yillik-hesaplama--87496594>
- <https://www.youtube.com/watch?v=M0ALc-Ka3oQ>