

# -RAPOR-

HAZIRLAYAN;

AD:Ali

SOYAD:DORAN

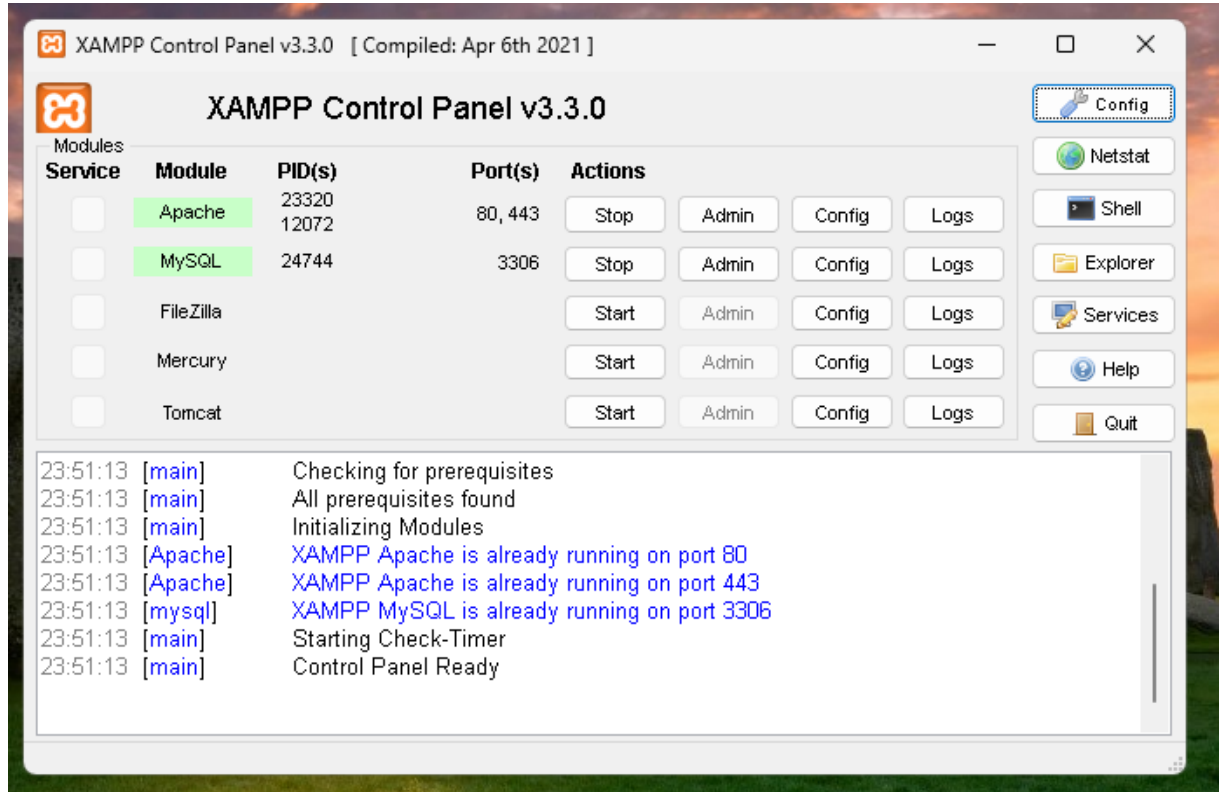
NO:222503004

İÇİNDEKİLER:

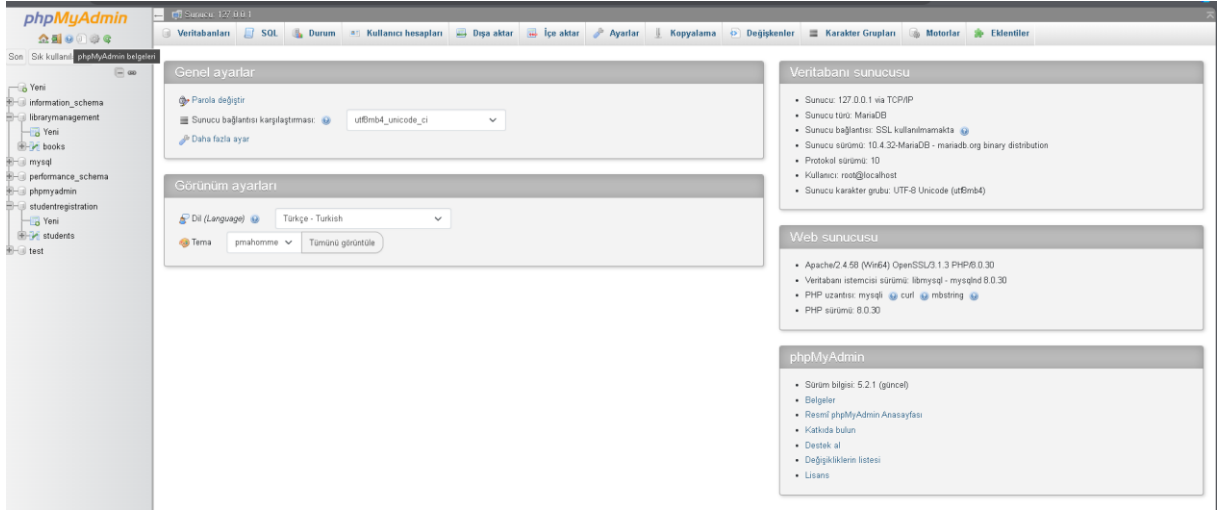
1. GENEL KURULUM VE YOL HARİTASI
2. KİTAPLIK YÖNETİM SİSTEMİ
3. ÖĞRENCİ KAYIT SİSTEMİ

## 1. GENEL KURULUM VE YOL HARİTASI :

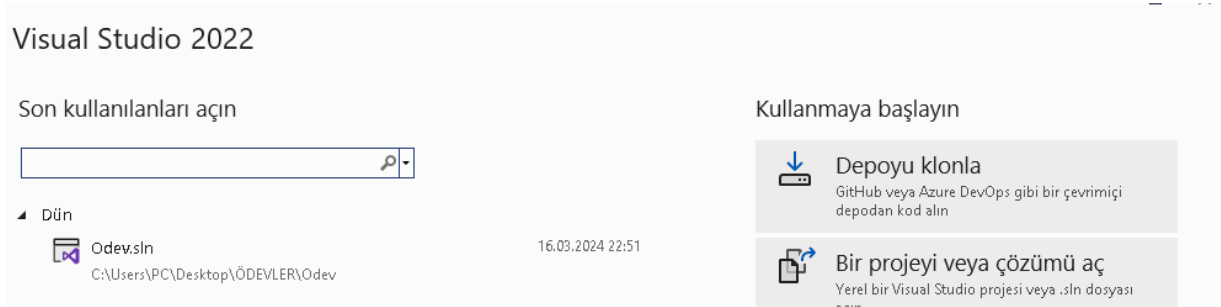
XAMPP uygulaması kurulur ve Apache ,MySQL başlatılır bu sayede phpmyadmin e bağlanılabilir.



Kullanıcı adı root ve şifre boş geçilerek phpmyadmin e giriş yapılır.



Tasarım için Visual Studio kurulur ve Odev için bir Windows Forms Framework projesi oluşturulur;

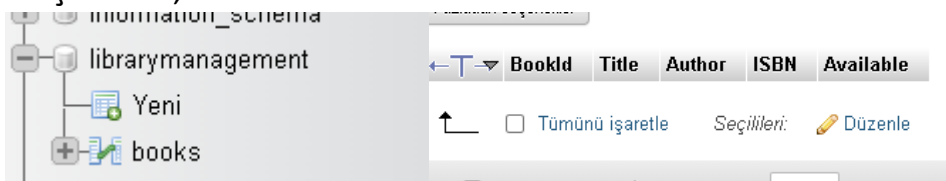


Artık proje yapım aşamasına hazırız ilk olarak Odev adında oluşturduğum forms uygulamasında kitaplık yönetim sistemi oluşturacağız.

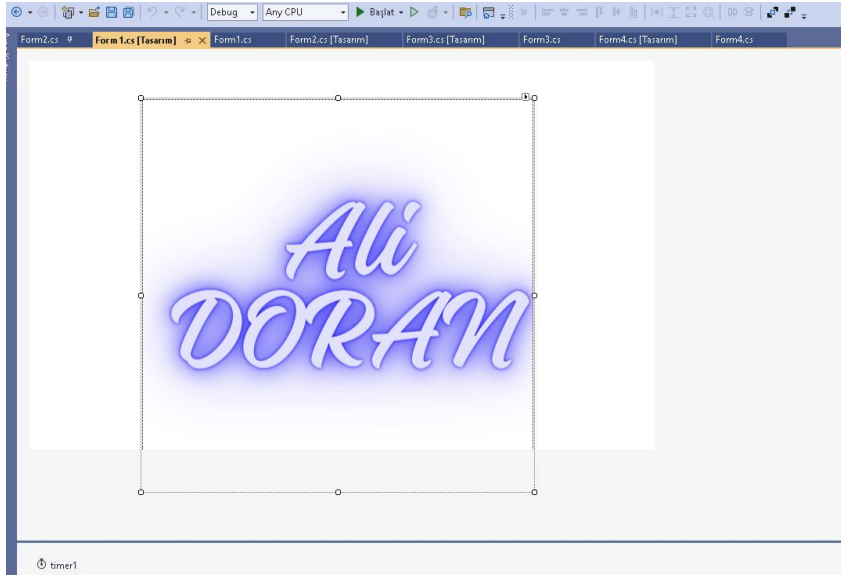
## 2. KİTAPLIK YÖNETİM SİSTEMİ :

**Amaç:** Kullanıcıların kitap eklemesine, silmesine ve mevcut kitapları listelemesine izin verecek bir arayüz oluşturma.

2.1.Verilen yönergeye göre belirtilen ad da veritabanı ve tablo oluşturulur;



2.2. **Form1 Tasarım** : Burada isim ve soyisim ile uygulamaya giriş yaparken bir giriş animasyonu oluşturulmuştur.



**Form1.cs** : Opaklık ayarını ayarlayıp güzel bir animasyon girişi tasarlanmıştır.

```
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 Odev
{
    //bu formda animasyonlu bir giriş yapılmıştır
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        bool islem = false;

        private void timer1_Tick(object sender, EventArgs e)
        {
            if (!islem) // Eğer islem false ise Opacity artırılmalı
            {
                this.Opacity += 0.009;
                if (this.Opacity >= 1.0) // Opacity 1.0 veya daha büyükse işlem tamamlandı, islem'i true yap
                {
                    islem = true;
                }
            }
            else // Eğer islem true ise Opacity azaltılmalı
            {
                this.Opacity -= 0.009;
                if (this.Opacity <= 0) // Opacity 0 veya daha küçükse yeni form açılmalı ve timer durdurulmalı
                {
                    Form2 gtr = new Form2(); //form2 ye direkt geç
                    gtr.Show();
                    timer1.Enabled = false;
                }
            }
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }
    }
}
```

2.3. **Form2 Tasarım** : Label ile kullanıcı bilgilendirilmiş ve buton aracı ile diğer formlara yönlendireme amaçlanmıştır.

KÜTÜPHANE YÖNETİM UYGULAMASINA HOŞGELDİNİZ :)

LÜTFEN BİR İŞLEM SEÇİNİZ;

KÜTÜPHANE VERİ TABANINA KİTAP EKLE :

VERİ TABANINDAN KİTAP LİSTELEME VE SİLME :

HAZIRLAYAN : Ali DORAN  
222503004

**Form2.cs** : button1\_Click ile diğer formlara yönlendirme işlemi tanımlanmıştır. this.Close() ile diğer formlara geçince önceki formun kapatılması(saklanması) sağlanmıştır. Eğer button1 e basılırsa form3 e ,button2 ye basılırsa form4 de gidecektir.

```
}  
  
1 başvuru  
private void button1_Click(object sender, EventArgs e)  
{  
    Form3 form3 = new Form3();  
    form3.Show();  
    this.Close();  
}  
  
1 başvuru  
private void button2_Click(object sender, EventArgs e)  
{  
    Form4 form4 = new Form4();  
    form4.Show();  
    this.Close();  
}  
  
1 başvuru
```

2.4. **Form3 Tasarım** : button1 yani kaydet butonu ile textboxlara girilen değerlerin sunucuya gönderilmesi amaçlanmıştır. Burda kitapid primarykey olup eğer aynı değerde iki kullanıcı kaydedilmeye çalışırsa hata verecektir. Anasayfa butnuna basılarak kayıt işleminden sonra form2 ye geri dönülebilir.

**Form3.cs** : öncelikle textboxlardaki verileri sunucuya göndereceğimiz için gerekli kütüphaneleri eklememiz gerekiyor.

```
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using MySql.Data.MySqlClient; // MySQL veritabanıyla etkileşim için gerekli kütüphane  
using System.Windows.Forms;  
using System.Data.SqlClient; // SQL Server veritabanıyla etkileşim için gerekli kütüphane
```

Button1 e basınca gerçekleşecek olayları sırayla açıklayalım;

Oluşturduğumuz veri tabanındaki sütunların hangi textboxa geleceğini belirtiyoruz. Eğer herhangi bir textbox boş geçilip butona basarsak uygulama hata verir . daha sonra veri tabanı bağlantısı için gerekli bilgileri “string connectionString=” ile giriyoruz ve veri tabanı ile ilişkilendirme kodlarımız yazıyoruz. Eğer herhangi bir hata yapmazsak “kitap başarıyla eklendi yazısıyla karşılaşıyoruz. Fakat eksik veya yanlış bir bağlantı yapmışsak hata alıyoruz.

```

1 baguru
private void button1_Click(object sender, EventArgs e) // "button1" butonuna tıklandığında gerçekleşecek olay
{
    // Form3'teki textBox'lerden verileri al
    string bookId = textBox1.Text;
    string title = textBox2.Text;
    string author = textBox3.Text;
    string isbn = textBox4.Text;

    // Herhangi bir textBox boş ise hata ver
    if (string.IsNullOrEmpty(bookId) || string.IsNullOrEmpty(title) || string.IsNullOrEmpty(author) || string.IsNullOrEmpty(isbn))
    {
        MessageBox.Show("Lütfen tüm alanları doldurun.", "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
        return; // Metodu burada sonlandır
    }

    // Veritabanına bağlantı için gerekli bilgiler
    string connectionString = "server=127.0.0.1;user=root;database=librarymanagement;password=";

    using (MySqlConnection connection = new MySqlConnection(connectionString))
    {
        // SQL komutu hazırla
        string query = "INSERT INTO books (BookId, Title, Author, ISBN, Available) VALUES (@BookId, @Title, @Author, @ISBN, @Available)";

        // Komutu ve bağlantıyı ilişkilendir
        using (MySqlCommand command = new MySqlCommand(query, connection))
        {
            // Parametreleri ekle
            command.Parameters.AddWithValue("@BookId", bookId);
            command.Parameters.AddWithValue("@Title", title);
            command.Parameters.AddWithValue("@Author", author);
            command.Parameters.AddWithValue("@ISBN", isbn);
            command.Parameters.AddWithValue("@Available", 1); // Yeni kitap eklendiği için Available değeri 1

            try
            {
                // Bağlantıyı aç
                connection.Open();

                // Komutu çalıştır
                int rowsAffected = command.ExecuteNonQuery();

                // İşlem başarılı mesajı
                MessageBox.Show("Kitap başarıyla eklendi.");

                // Form3'ü kapat
            }
            catch (Exception ex)
            {
                // Hata durumunda mesaj göster
                MessageBox.Show("Hata oluştu: " + ex.Message, "Hata", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
        }
    }
}

1 baguru
private void button2_Click(object sender, EventArgs e)
{
    Form2 form2 = new Form2();
    form2.Show();

    // Form3'ü kapat
    this.Close();
}

```

2.5. **Form4 Tasarım** : sunucudaki veriyi formda listelemek için **dataGridView1** adında bir **dataGridView** ekledik ve eğer sunucudan bir veri silmek istiyorsak listelenmiş bir kitabı seçip sil butonuna basınca artık hem listemizden hem de sunucumuzdan kitap bilgilerini silebiliyoruz. Ayrıca ana menüye dön butonu ile form2 ye geri dönebiliriz.

Form4.cs :

```
5 başvuru
public partial class Form4 : Form
{
    MySqlConnection connection; // MySQL bağlantısı için değişken
    MySqlCommand command; // MySQL komutları için değişken
    MySqlDataAdapter adapter; // MySQL veri okuma ve yazma işlemleri için değişken
    DataTable table; // Verilerin saklanacağı tablo

    1 başvuru
    public Form4()
    {
        InitializeComponent();
        dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect; // Tüm satırın seçilmesini sağlar
        dataGridView1.AllowUserToAddRows = false; // Kullanıcının yeni satır eklemesini engeller
        dataGridView1.ReadOnly = true; // Verilerin sadece okunabilir olmasını sağlar
        dataGridView1.AutoGenerateColumns = true; // Sütunları otomatik oluşturur
        dataGridView1.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill; // Sütunları otomatik olarak genişletir
        LoadData(); // Veritabanından verileri yükler
    }

    1 başvuru
    private void LoadData()
    {
        string connectionString = "server=127.0.0.1;user=root;database=librarymanagement;password="; // MySQL bağlantı dizesi
        connection = new MySqlConnection(connectionString); // Bağlantı nesnesi oluşturulur ve açılır
        connection.Open();
        command = new MySqlCommand("SELECT * FROM books", connection); // Veritabanından tüm kitapları seçen sorgu
        adapter = new MySqlDataAdapter(command); // Verileri almak için veri adaptörü oluşturulur
        table = new DataTable(); // Verilerin saklanacağı tablo oluşturulur
        adapter.Fill(table); // Veriler tabloya doldurulur
        dataGridView1.DataSource = table; // DataGridView kontrolüne veriler atanır
    }

    1 başvuru
    private void button1_Click(object sender, EventArgs e)
    {
        // DataGridView'de bir satırın seçilip seçilmediği kontrol edilir
        if (dataGridView1.SelectedRows.Count > 0)
        {
            // Kullanıcıya silme işlemi için onay istenir
            DialogResult result = MessageBox.Show("Seçili satırı silmek istediğinizden emin misiniz?", "Uyarı", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);
            if (result == DialogResult.Yes)
            {
                int selectedRowIndex = dataGridView1.SelectedRows[0].Index; // Seçilen satırın indeksi alınır
                string selectedBookId = dataGridView1.Rows[selectedRowIndex].Cells["BookId"].Value.ToString(); // Seçilen satırın kitap ID'si alınır

                string deleteQuery = "DELETE FROM books WHERE BookId = @BookId"; // Kitap silme sorgusu
                command = new MySqlCommand(deleteQuery, connection); // Komut oluşturulur
                command.Parameters.AddWithValue("@BookId", selectedBookId); // Parametre eklenir
                command.ExecuteNonQuery(); // Sorgu çalıştırılır

                table.Rows.RemoveAt(selectedRowIndex); // Tablodan seçilen satır silinir
                MessageBox.Show("Seçilen kitap başarıyla silindi."); // Kullanıcıya mesaj gösterilir
            }
        }
        else
        {
            MessageBox.Show("Lütfen bir satır seçiniz."); // Kullanıcıya bir satır seçmesi gerektiği hakkında bilgi verilir
        }
    }

    1 başvuru
    private void button2_Click(object sender, EventArgs e)
    {
        // Yeni kitap ekleme formunu açar ve mevcut formu kapatır
        Form2 form2 = new Form2();
        form2.Show();
    }
}
```

## Ödev1 Örnek :

ÖDEVLER Yeni Metin

Form3

**LÜTFEN EKLEMELİ İSTEDİĞİNİZ KİTAPIN ID ,BAŞLIK , YAZAR , ISBN BİLGİLERİNİ GİRİN :**

KİTAP ID : 1

BAŞLIK : suç ve ceza

YAZAR : dostoveyski

ISBN : 1234

KAYDET

Ana Sayfaya Dön

ÖDEVLER Yeni Metin

Form3

**LÜTFEN EKLEMELİ İSTEDİĞİNİZ KİTAPIN ID ,BAŞLIK , YAZAR , ISBN BİLGİLERİNİ GİRİN :**

KİTAP ID : 1

BAŞLIK : suç ve ceza

YAZAR : dostoveyski

ISBN : 1234

Kitap başarıyla eklendi.

Tamam

KAYDET

Ana Sayfaya Dön



Form4

ÖDEVLER Yeni Metin

VERİTABANINDAKİ KİTAPLAR BURADA LİSTELENMİŞTİR LÜTFEN SİLMEK İSTEDİĞİNİZ KİTABI SEÇİN VE SİL BUTONUNA BASINIZ :

	BookId	Title	Author	ISBN	Availa
▶	1	suç ...	dost...	1234	1

SİL

Ana Sayfaya Dön

Fazladan seçenekler

← T →

	BookId	Title	Author	ISBN	Available
<input type="checkbox"/>  Düzenle  Kopyala  Sil	1	suç ve ceza	dostoveyski	1234	1

↑ ☐ Tümünü işaretle Seçileri:  Düzenle  Kopyala  Sil  Dışa aktar

Form4

ÖDEVLER Yeni Metin

VERİTABANINDAKİ KİTAPLAR BURADA LİSTELENMİŞTİR LÜTFEN SİLMEK İSTEDİĞİNİZ KİTABI SEÇİN VE SİL BUTONUNA BASINIZ :

	BookId	Title	Author	ISBN	Availa
▶	1	suç ...	dost...	1234	1

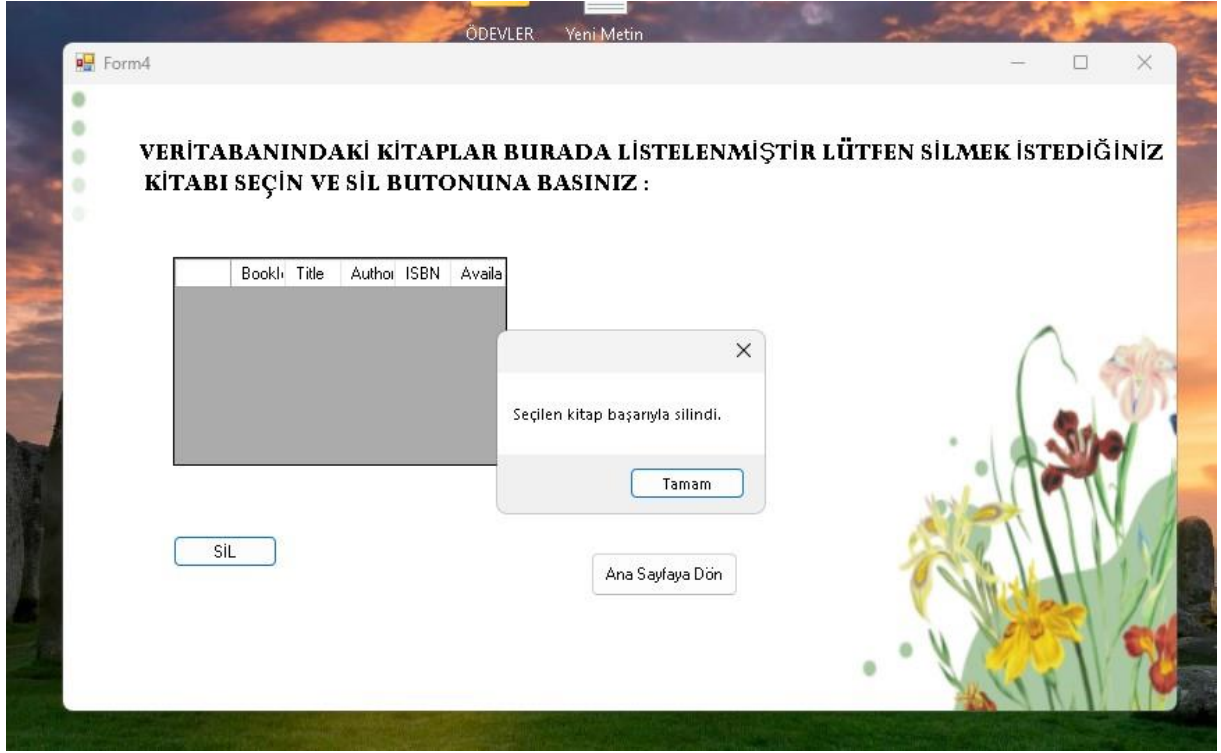
SİL

Ana Sayfaya Dön

Uyarı

Seçili satırı silmek istediğinizden emin misiniz?

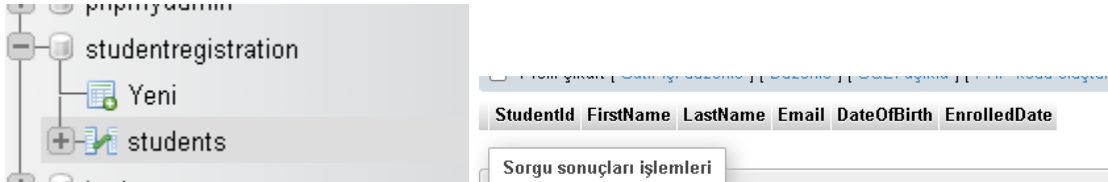
Evet Hayır



### 3. ÖĞRENCİ KAYIT SİSTEMİ:

**Amaç:** öğrenci ekleyebileceğimiz ve listeleyebileceğimiz bir öğrenci kayıt sistemi oluşturmak.

Phpmyadminde belirtilen adda veri tabanı ve tablo oluştur;



3.1. **Form1 Tasarım:** ödev1 de olduğu gibi yine aynı animasyonlu giriş yapılmıştır.



## Form1.cs:

```
1 başvuru
public Form1()
{
    InitializeComponent();
}

bool islem = false;
1 başvuru
private void timer1_Tick(object sender, EventArgs e)
{
    if (!islem) // Eğer islem false ise Opacity artırılmalı
    {
        this.Opacity += 0.009;
        if (this.Opacity >= 1.0) // Opacity 1.0 veya daha büyükse işlem tamamlandı, islem'i true yap
        {
            islem = true;
        }
    }
    else // Eğer islem true ise Opacity azaltılmalı
    {
        this.Opacity -= 0.009;
        if (this.Opacity <= 0) // Opacity 0 veya daha küçükse yeni form açılmalı ve timer durdurulmalı
        {
            Form2 gtr = new Form2(); // form2 ye direkt geç
            gtr.Show();
            timer1.Enabled = false;
        }
    }
}
```

- 3.2. **Form2 Tasarım :** kayıt butonuna basarak textboxlar ile kullanıcıdan öğrenci bilgilerini ,dateTimerPicker ile tarih bilgilerini alıp sunucuya iletmek amaçlanmıştır.



**Form2.cs:** öncelikle sql kütüphanesi eklenir ve sunucu ile bağlantı bilgileri girilerek hangi girdilerin tablodaki hangi sütunlarla eşleştiği bilgileri verilir . eğer herhangi bir hata yapılmadıysa veri başarıyla

eklendi yazısı çıkar ve Form3 e yönlendirerek burada listeleme işlemi gerçekleşir.

```
using System;
using System.Threading.Tasks;
using MySql.Data.MySqlClient;
using System.Data.SqlClient;
using System.Windows.Forms;

namespace Odev2
{
    7 başlatma
    public partial class Form2 : Form
    {
        2 başlatma
        public Form2()
        {
            InitializeComponent();
        }

        1 başlatma
        private void button1_Click(object sender, EventArgs e)
        {
            string connectionString = "server=127.0.0.1;user=root;database=studentregistration;password=";
            using (MySqlConnection connection = new MySqlConnection(connectionString))
            {
                string query = "INSERT INTO Students (StudentId, FirstName, LastName, Email, DateOfBirth, EnrolledDate) VALUES (@StudentId, @FirstName, @LastName, @Email, @DateOfBirth, @EnrolledDate)";
                using (MySqlCommand command = new MySqlCommand(query, connection))
                {
                    command.Parameters.AddWithValue("@StudentId", textBox1.Text);
                    command.Parameters.AddWithValue("@FirstName", textBox2.Text);
                    command.Parameters.AddWithValue("@LastName", textBox3.Text);
                    command.Parameters.AddWithValue("@Email", textBox4.Text);
                    command.Parameters.AddWithValue("@DateOfBirth", dateTimePicker1.Value.ToString("yyyy-MM-dd"));
                    command.Parameters.AddWithValue("@EnrolledDate", dateTimePicker2.Value.ToString("yyyy-MM-dd"));

                    try
                    {
                        connection.Open();
                        command.ExecuteNonQuery();
                        MessageBox.Show("Veri başarıyla eklendi.");

                        // Veriler başarıyla eklendiğinden emin ol
                        // Form3'e geçiş yap
                        Form3 form3 = new Form3();
                        form3.LoadDataFromDatabase(); // Form3'teki DataGridView'e verileri yükle
                        form3.Show();

                        // Form2'yi kapat
                        this.Hide(); // Form2'nin görünürliğini kaldırarak gizler
                    }
                    catch (Exception ex)
                    {
                        MessageBox.Show("Veri eklenirken bir hata oluştu: " + ex.Message);
                    }
                }
            }
        }
    }
}
```

3.3. **Form3 Tasarım :** burada bulunan dataGridView ile sunucudan alınan veriler listelenir . ve buton ile form2 ye geri dönülebilir.

Form3.cs :

```

namespace Odev2
{
    5 bayyuru
    public partial class Form3 : Form
    {
        1 bayyuru
        public Form3()
        {
            InitializeComponent();

            // Veritabanından verileri yükleme işlevi
            1 bayyuru
            public void LoadDataFromDatabase()
            {
                // Veritabanı bağlantı dizesi
                string connectionString = "server=127.0.0.1;user=root;database=studentregistration;password=";

                // Sorgu metni
                string query = "SELECT * FROM Students";

                // MySqlConnection kullanarak bağlanılır ve bağlantıyı otomatik olarak kapatır
                using (MySqlConnection connection = new MySqlConnection(connectionString))
                {
                    using (MySqlCommand command = new MySqlCommand(query, connection))
                    {
                        try
                        {
                            // Bağlantıyı aç
                            connection.Open();

                            // Veri adaptörü oluştur
                            MySqlDataAdapter adapter = new MySqlDataAdapter(command);

                            // Verileri tutacak bir DataTable oluştur
                            DataTable table = new DataTable();

                            // Veri adaptörüyle DataTable'ı doldur
                            adapter.Fill(table);

                            // DataGridView kontrolüne DataTable'ı bağla
                            dataGridView1.DataSource = table;
                        }
                        catch (Exception ex)
                        {
                            // Hata durumunda kullanıcıya bilgi ver
                            MessageBox.Show("Veri çekme işleminde bir hata oluştu: " + ex.Message);
                        }
                    }
                }
            }

            // Diğer formu açmak için kullanılan düğme tıklayıcı işlevi
            1 bayyuru
            private void button1_Click(object sender, EventArgs e)
            {
                // Form2'yi göster
                Form2 form2 = new Form2();
                form2.Show();

                // Form3'ü kapat
                this.Close();
            }
        }
    }
}

```

## Ödev2 Örnek :

ÖĞRENCİ KAYIT SİSTEMİNE HOŞ GELDİNİZ LÜTFEN AŞAĞIDAKİ BİLGİLERİ DOLDURUNUZ :

ÖĞRENCİ NO : 222503004

ADI : Ali

SOYADI : Doran

EMAIL : ali@gmail.com

DOĞUM TARİHİ : 17 Mart 2024 Pazar

KAYIT TARİHİ : 17 Mart 2024 Pazar

HAZIRLAYAN : Ali DORAN

222503004

KAYIT

ÖĞRENCİ KAYIT SİSTEMİNE HOŞ GELDİNİZ LÜTFEN AŞAĞIDAKİ BİLGİLERİ DOLDURUNUZ :

ÖĞRENCİ NO : 222503004

ADI : Ali

SOYADI : Doran

EMAIL : ali@gmail.com

DOĞUM TARİHİ : 17 Mart 2024 Pazar

KAYIT TARİHİ : 17 Mart 2024 Pazar

HAZIRLAYAN : Ali DORAN

222503004

KAYIT

Veri başarıyla eklendi.

Tamam



ÖDEVLER Yeni Metin

Form3

KAYITLI ÖĞRENCİ LİSTESİ :

	StudentId	FirstName	LastName	Email	DateOfBirth
▶	222503004	Ali	Doran	ali@gmail.com	17.03.2024
*					

KAYIT EKRANINA DÖN

Fazladan seçenekler