

-RAPOR-

HAZIRLAYAN;

AD SOYAD: Ali DORAN

NO: 222503004

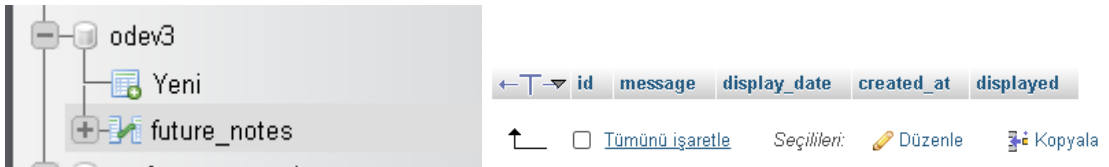
İçindekiler;

- 1- **GİRİŞ** : Proje hakkında ön bilgi ve yapılmak istenilen hakkında bilgi verme
- 2- **SUNUCU BAĞLATISI**: Gerekli veri tabanı ve tablo oluşturulur.
- 3- **PROJE TANITIM**: Form tasarım ve form.cs resimleri ile projeyi detaylı anlatım.
 - a- Form1 tasarım/Form1.cs
 - b- Form2 tasarım/Form2.cs
 - c- Form3 tasarım/Form3.cs
 - d- Form4 tasarım/Form4.cs
 - e- Form5 tasarım/Form5.cs

1- **GİRİŞ** : Kullanıcıların belirli bir tarihte okunmak üzere notlar bırakmalarını sağlayacak bir Windows Forms uygulaması tasarlamak amaçlanmıştır. Bana özel olarak her projemde koyduğum, proje başlatıldığında isim ve soyismimin bulunduğu bir animasyonlu giriş ekranı sizi karşılayacaktır. Ayrıca modern bir tasarım amaçlandığı için butonların simgeleri değiştirilmiş ayrıca MouseEnter ve MouseLeave olaylarını EventHandler ile ilişkilendirerek fare buton üzerine geldiğinde renk değişimi ve üzerinden çıkarıldığında eski haline gelme gibi tasarımsal incelikler yapılmıştır. Ayrıca ilk formumuz olan form1 de hazırlayanın imazı olan isim ve soyisim label aracılığı ile yazılmıştır.

Not: Ayrıca uygulama farklılığı amacıyla sunucu tarafında ekstra olarak “displayed” sütunu eklenerek eğer notların gösterilme zamanı gelmiş ise ve kullanıcı bunu görmüş ise displayed sütunu 0 dan 1 e dönerek gösterilen formdan kaldırılarak farklı bir formda bulunan ve geçmiş notların bulunduğu listeye eklenmesi sağlanıyor (zaten formlar kısmında detaylandırılacaktır).

2- **SUNUCU BAĞLATISI**: Yönergeye uygun veri tabanı ve tablo oluşturulur. Bu sayede uygulama üzerinden gelen veriler burada depolanabilir ve istenildiğinde çağrılabilir. Yönergeye ek olarak notların görülüp görülmediğini kontrol edebilmek için “displayed” adında ve sadece 1 ve 0 değerlerini alabilen bir sütun daha eklenmiştir.

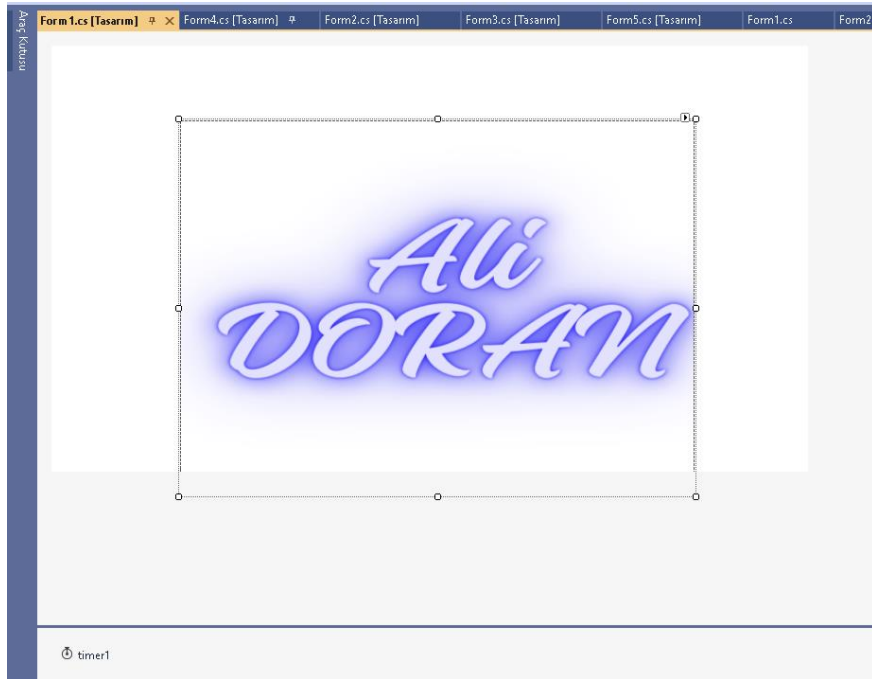


Gerekli kütüphane ve alttaki string yapısı ile uygulama ve sunucu arasındaki bağlantı sağlanmış olur. Tabi ki kontrol panelinden “apache” ve “mysql” start edilmesi unutulmaması gerekir.

```
// Veritabanı bağlantı dizesi.  
string connectionString = "server=127.0.0.1;user=root;database=odev3;password=";  
using (SqlConnection conn = new SqlConnection(connectionString))
```

3- PROJE TANITIM:

a- **Form1 Tasarım:** Animasyonlu bir giriş için isim ve soyismimin yazılı olduğu bir picturebox ve timer aracı ile güzel bir animasyonlu giriş tasarladık.



Form1.cs : opaklık ayarlarıyla oynanıp görselin giderek kaybolması sağlanır ve sonraki forma geçilmesi sağlanır

```

namespace Odev3
{
    4 bagvuru
    public partial class Form1 : Form
    {
        1 bagvuru
        public Form1()
        {
            InitializeComponent();
        }

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

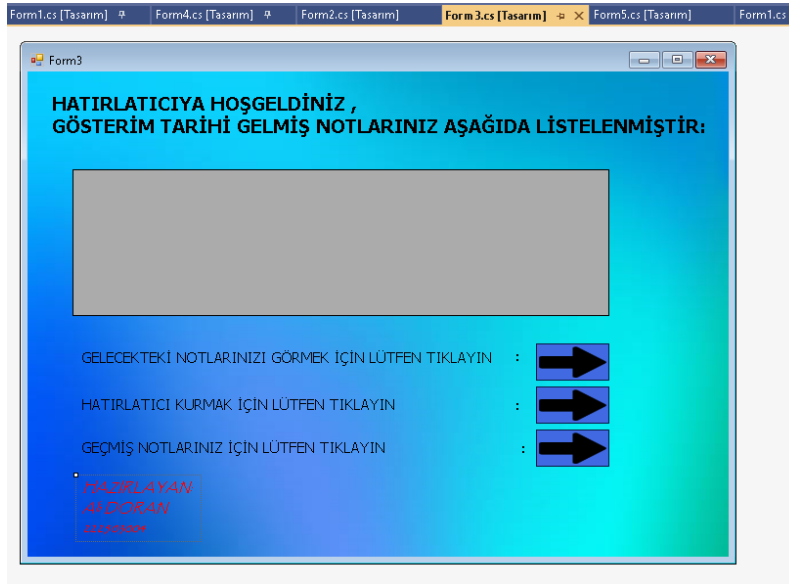
        1 bagvuru
        private void pictureBox1_Click(object sender, EventArgs e)
        {
        }

        bool islem = false;
        1 bagvuru
        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 <= 0.009) // Opaklık 0'a düştüğünde yeni form açılmadan önce kontrol yap
                {
                    timer1.Enabled = false; // Timer'ı durdur
                    Form3 gtr = new Form3(); // Form3 nesnesi oluştur
                    gtr.Show(); // Form3'ü göster
                    this.Hide(); // Form1'i gizle veya kapat
                }
            }
        }
    }
}

```

Not: Form uygulamamızın girişi Form3 olduğu için Form3 ten devam ediliyorum.

- c- **Form3 tasarım:** Uygulamamızın ana sayfasıdır. Eğer önceden hatırlatıcı kurduysak ve zamanı gelmiş ise veya zamanı geçmiş fakat biz bu notu hala görmemişsek burada listelenir. Ayrıca burada bulunan label yönlendirmelerinin karşısındaki butonlara basılarak yapmak istediğimiz işlemleri gerçekleştirmek için diğer formlara gidebiliyoruz. Kısacası burada “display=0” ve zamanı gelen not varsa listelenir.



Form3.cs: Butonlar için üzerine gelme ve ayrılma durumlarını yapmak için laylar tanımlanmış. Ayrıca veri tabanı ile bağlantı sağlanmış , form yüklenince gerekli şartları sağlıyorsa notların listelenmesi sağlanmıştır.

```

4 basvuru
public Form3()
{
    InitializeComponent();
    // Aşağıdaki satırlar, button1, button2 ve button3 için MouseEnter ve MouseLeave olaylarını tanımlar.
    // Bu olaylar, mouse butonların üzerine geldiğinde veya çıktığında tetiklenir.
    this.button1.MouseEnter += new EventHandler(button1_MouseEnter);
    this.button1.MouseLeave += new EventHandler(button1_MouseLeave);
    this.button2.MouseEnter += new EventHandler(button2_MouseEnter);
    this.button2.MouseLeave += new EventHandler(button2_MouseLeave);
    this.button3.MouseEnter += new EventHandler(button3_MouseEnter);
    this.button3.MouseLeave += new EventHandler(button3_MouseLeave);
}

// Form3 yüklendiğinde çalışacak metod. Görüntülenmemiş hatırlatıcıları yükler.
1 basvuru
private void Form3_Load(object sender, EventArgs e)
{
    LoadReminders(); // Hatırlatıcıları yükleyen metodu çağırır.
}

// Veritabanından hatırlatıcıları yükleyip dataGridView1 kontrolünde listeler.
1 basvuru
private void LoadReminders()
{
    // Veritabanı bağlantı dizesi.
    string connectionString = "server=127.0.0.1;user=root;database=odev3;password=";
    using (MySqlConnection conn = new MySqlConnection(connectionString))
    {
        try
        {
            conn.Open(); // Veritabanı bağlantısını açar.
            // Geçmişe dönük ve henüz gösterilmemiş hatırlatıcıları seçmek için SQL sorgusu.
            string sqlSelect = "SELECT * FROM future_notes WHERE display_date <= NOW() AND displayed = 0";
            MySqlCommand cmdSelect = new MySqlCommand(sqlSelect, conn);
            MySqlDataAdapter adapter = new MySqlDataAdapter(cmdSelect);
            DataTable dt = new DataTable();
            adapter.Fill(dt); // Sonuçları DataTable'a doldurur.

            dataGridView1.DataSource = dt; // Sonuçları dataGridView1 kontrolünde gösterir.

            // Eğer sonuç varsa, bu notların gösterildiği olarak işaretlenmesi.
            if (dt.Rows.Count > 0)
            {
                string sqlUpdate = "UPDATE future_notes SET displayed = 1 WHERE display_date <= NOW() AND displayed = 0";
                MySqlCommand cmdUpdate = new MySqlCommand(sqlUpdate, conn);
                cmdUpdate.ExecuteNonQuery();
            }
        }
        catch (Exception ex) // Bir hata oluşursa kullanıcıya hata mesajı gösterir.
        {
            MessageBox.Show("Bir hata oluştu: " + ex.Message);
        }
    }
}
}

```

Burada da butonlar için üzerine gelme veya ayrılma durumunda ve butonlara basınca yapılacak durumlar tanımlanmıştır.

```

// button1, button2 ve button3 için MouseEnter ve MouseLeave olaylarını işleyen metodlar.
// Bu metodlar, butonların arka plan rengini değiştirerek görsel bir geri bildirim sağlar.
1 basvuru
private void button1_MouseEnter(object sender, EventArgs e) { button1.BackColor = ControlPaint.Light(Color.RoyalBlue); }
1 basvuru
private void button1_MouseLeave(object sender, EventArgs e) { button1.BackColor = Color.RoyalBlue; }
1 basvuru
private void button2_MouseEnter(object sender, EventArgs e) { button2.BackColor = ControlPaint.Light(Color.RoyalBlue); }
1 basvuru
private void button2_MouseLeave(object sender, EventArgs e) { button2.BackColor = Color.RoyalBlue; }
1 basvuru
private void button3_MouseEnter(object sender, EventArgs e) { button3.BackColor = ControlPaint.Light(Color.RoyalBlue); }
1 basvuru
private void button3_MouseLeave(object sender, EventArgs e) { button3.BackColor = Color.RoyalBlue; }

// button2 ve button3 tıklandığında çalışacak metodlar. Yeni formlar açar ve mevcut formu kapatır.
1 basvuru
private void button2_Click(object sender, EventArgs e) { Form2 form2 = new Form2(); form2.Show(); this.Close(); }
1 basvuru
private void button3_Click(object sender, EventArgs e) { Form5 form5 = new Form5(); form5.Show(); this.Close(); }

// button1 tıklandığında çalışacak metod. Yeni bir form açar ve mevcut formu kapatır.
1 basvuru
private void button1_Click(object sender, EventArgs e) { Form4 form4 = new Form4(); form4.Show(); this.Close(); }

```

b- Form2 Tasarım: Bu alanda kullanıcının not girmesi için bir textbox , tarih ve zamanı girebilmesi için dateTimePicker aracı vardır. Kaydet butonu yardımıyla bu alanlara girilen veriler sunucuya gönderilir. Ana sayfa butonuna basılarak da giriş sayfası olan form 3 e geri dönülebilir.

Form2.cs: yine butonlar için üzerine gelme özelliği tanımlanır. Sunucu bağlantısıyla beraber buradan alınan bilgiler sunucuya kaydedilir. Eğer işlemde herhangi bir hata olursa notunuz kaydedildi diye uyarı mesajı yansıtır . herhangi bir hata varsada o hatayı döndürür.

```

namespace Odev3
{
    5 satırı
    public partial class Form2 : Form
    {
        1 satırı
        public Form2()
        {
            InitializeComponent();
            // Aşağıdaki satırlar, button1 ve button2 için MouseEnter ve MouseLeave olaylarını tanımlar.
            // Bu olaylar, mouse butonun üzerine geldiğinde veya çıktığında tetiklenir.
            this.button1.MouseEnter += new EventHandler(button1_MouseEnter);
            this.button1.MouseLeave += new EventHandler(button1_MouseLeave);
            this.button2.MouseEnter += new EventHandler(button2_MouseEnter);
            this.button2.MouseLeave += new EventHandler(button2_MouseLeave);
        }

        // button1 tıklandığında çalışacak metod. Bu metod, kullanıcıdan alınan notu ve tarihi bir veritabanına kaydeder.
        1 satırı
        private void button1_Click(object sender, EventArgs e)
        {
            // textBox1'den alınan metni ve dateTimePicker1'den alınan tarihi değişkenlere atar.
            string message = textBox1.Text;
            DateTime displayDate = dateTimePicker1.Value;

            // Veritabanı bağlantı dizesi. Kullanıcı adı, sunucu, veritabanı adı belirtilir.
            string connectionString = "server=127.0.0.1;user=root;database=odev3;password=";
            using (MySQLConnection conn = new MySqlConnection(connectionString))
            {
                try
                {
                    conn.Open(); // Veritabanı bağlantısını açar.
                    // Veritabanına not eklemek için SQL komutu. @message ve @displayDate parametreleri kullanılır.
                    string sqlInsert = "INSERT INTO future_notes (message, display_date, created_at, displayed) VALUES (@message, @displayDate, NOW(), 0)";
                    MySqlCommand cmdInsert = new MySqlCommand(sqlInsert, conn);
                    // Parametrelere değerler atanır.
                    cmdInsert.Parameters.AddWithValue("@message", message);
                    cmdInsert.Parameters.AddWithValue("@displayDate", displayDate);

                    cmdInsert.ExecuteNonQuery(); // SQL komutunu çalıştırır.
                    MessageBox.Show("Notunuz kaydedildi."); // Başarılı bir kayıt sonrası kullanıcıya bilgi verir.
                }
                catch (Exception ex) // Bir hata olursa kullanıcıya hata mesajı gösterir.
                {
                    MessageBox.Show("Bir hata oluştu: " + ex.Message);
                }
            }
        }
    }
}

```

```

// button1 üzerine mouse ile gelindiğinde arka plan rengini açık mavi yapar.
1 başvuru
private void button1_MouseEnter(object sender, EventArgs e)
{
    button1.BackColor = ControlPaint.Light(Color.RoyalBlue);
}

// button1 üzerinden mouse ayrıldığında arka plan rengini orijinaline döndürür.
1 başvuru
private void button1_MouseLeave(object sender, EventArgs e)
{
    button1.BackColor = Color.RoyalBlue;
}

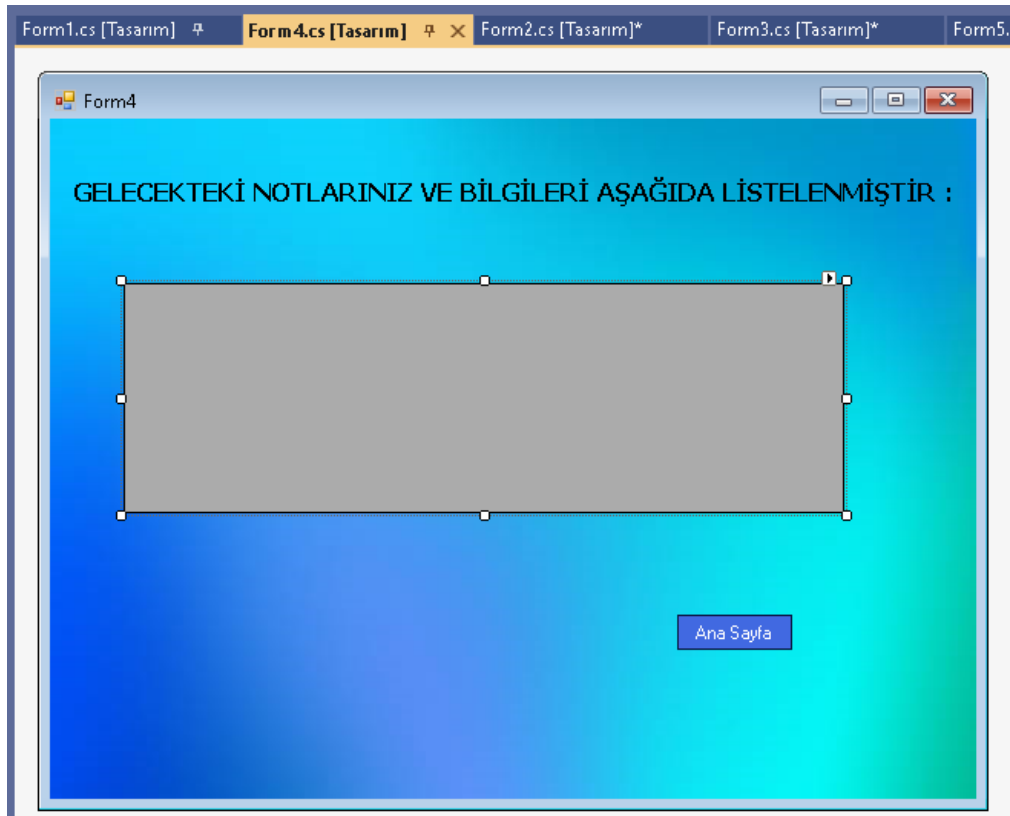
1 başvuru
private void button2_MouseEnter(object sender, EventArgs e)
{
    button2.BackColor = ControlPaint.Light(Color.RoyalBlue);
}

1 başvuru
private void button2_MouseLeave(object sender, EventArgs e)
{
    button2.BackColor = Color.RoyalBlue;
}

// button2 tıklandığında çalışacak metod. Form3'ü açar ve Form2'yi gizler.
1 başvuru
private void button2_Click(object sender, EventArgs e)
{
    Form3 form3 = new Form3(); // Form3 nesnesi oluşturur.
    form3.Show(); // Form3'ü gösterir.
    this.Hide(); // Mevcut formu (Form2) gizler.
}

```

d- Form4 Tasarım: Form3 teki buton ile form4 de gelip henüz zamanı gelmeyen notları burada listelenmiş şekilde görebiliriz. Notun zamanı gelince bu listeden silinip form3 te bulunan listede listelenir. Yine buton yardımı ile ana sayfaya dönebiliriz.



Form4.cs: yine sunucu bağlantısı ve mouse hareketleri için gerekli tanımlamalar yapılmış.

```
namespace Odev3
{
    5 bayyuru
    public partial class Form4 : Form
    {
        1 bayyuru
        public Form4()
        {
            InitializeComponent();
            // button1 için MouseEnter ve MouseLeave olaylarını EventHandler ile ilişkilendirir.
            this.button1.MouseEnter += new EventHandler(button1_MouseEnter);
            this.button1.MouseLeave += new EventHandler(button1_MouseLeave);
        }

        // Form yüklendiğinde çalışacak olan metod. Gelecekteki hatırlatıcıları yükler.
        1 bayyuru
        private void Form4_Load(object sender, EventArgs e)
        {
            LoadFutureReminders(); // Gelecekteki hatırlatıcıları yükleyen metodu çağırır.
        }

        // Gelecekte gösterilecek hatırlatıcıları veritabanından yükleyip, DataGridView kontrolünde listeler..
        1 bayyuru
        private void LoadFutureReminders()
        {
            // Veritabanı bağlantı dizesi.
            string connectionString = "server=127.0.0.1;user=root;database=odev3;password="; // Şifrenizi buraya ekleyin.
            using (MySqlConnection conn = new MySqlConnection(connectionString))
            {
                try
                {
                    conn.Open(); // Veritabanı bağlantısını açar.
                    // Gelecekteki tarihe sahip notları seçmek için SQL sorgusu.
                    string sqlSelect = "SELECT * FROM future_notes WHERE display_date > NOW()";
                    MySqlCommand cmdSelect = new MySqlCommand(sqlSelect, conn);
                    MySqlDataAdapter adapter = new MySqlDataAdapter(cmdSelect);
                    DataTable dt = new DataTable();
                    adapter.Fill(dt); // Sonuçları DataTable'a doldurur.

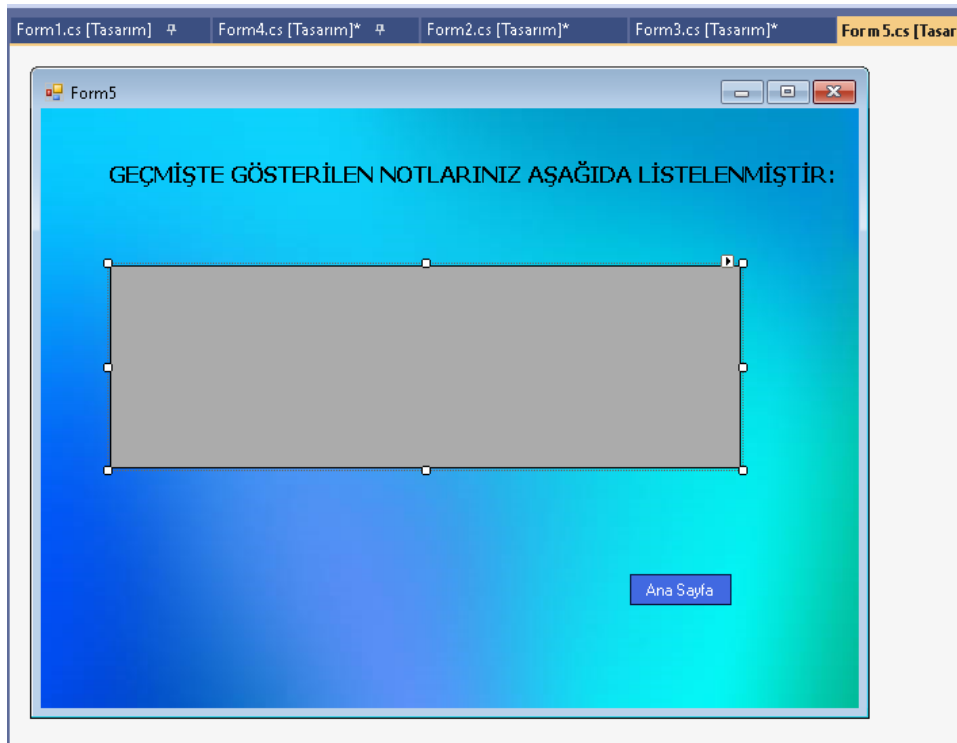
                    dataGridView1.DataSource = dt; // Sonuçları dataGridView1 kontrolünde gösterir.
                }
                catch (Exception ex) // Bir hata oluşursa kullanıcıya hata mesajı gösterir.
                {
                    MessageBox.Show("Bir hata oluştu: " + ex.Message);
                }
            }
        }

        // button1 üzerine mouse ile gelindiğinde arka plan rengini açık maviye çevirir.
        1 bayyuru
        private void button1_MouseEnter(object sender, EventArgs e)
        {
            button1.BackColor = ControlPaint.Light(Color.RoyalBlue);
        }

        // button1 üzerinden mouse ayrıldığında arka plan rengini orijinaline döndürür.
        1 bayyuru
        private void button1_MouseLeave(object sender, EventArgs e)
        {
            button1.BackColor = Color.RoyalBlue;
        }

        // button1 tıklandığında çalışacak olan metod. Form3'ü açar ve mevcut formu (Form4) kapatır.
        1 bayyuru
        private void button1_Click(object sender, EventArgs e)
        {
            Form3 form3 = new Form3(); // Form3'ün bir örneğini oluşturur.
            form3.Show(); // Form3'ü gösterir.
            this.Close(); // Mevcut formu (Form4) kapatır.
        }
    }
}
```

e-Form5 Tasarım: form3 ün yünlendirmesiyle buraya gelebilir ve şimdiye kadarki tüm geçmiş ve gördüğünüz notlarınız burada listelenir. Buton yardımıyla ana sayfaya dönülebilir.



Form5.cs: Form4 ile hemen hemen aynı şeyler sadece koşul ifadesi farklı şekilde.

```
namespace Odev3
{
    5 başvuru
    public partial class Form5 : Form
    {
        1 başvuru
        public Form5()
        {
            InitializeComponent();
            // button1 için MouseEnter ve MouseLeave olaylarını EventHandler ile ilişkilendirir.
            this.button1.MouseEnter += new EventHandler(button1_MouseEnter);
            this.button1.MouseLeave += new EventHandler(button1_MouseLeave);
        }

        // Form yüklendiğinde çalışacak olan metod. Geçmişte gösterilmiş hatırlatıcıları yükler.
        1 başvuru
        private void Form5_Load(object sender, EventArgs e)
        {
            LoadPastReminders(); // Geçmiş hatırlatıcıları yükleyen metodu çağırır.
        }

        // Geçmişte gösterilmiş hatırlatıcıları veritabanından yükleyip, DataGridView kontrolünde listeler.
        1 başvuru
        private void LoadPastReminders()
        {
            // Veritabanı bağlantı dizesi.
            string connectionString = "server=127.0.0.1;user=root;database=odev3;password=";
            using (MySQLConnection conn = new MySqlConnection(connectionString))
            {
                try
                {
                    conn.Open(); // Veritabanı bağlantısını açar.
                    // Daha önce gösterilmiş olan notları seçmek için SQL sorgusu.
                    string sqlSelect = "SELECT * FROM future_notes WHERE displayed = 1";
                    MySqlCommand cmdSelect = new MySqlCommand(sqlSelect, conn);
                    MySqlDataAdapter adapter = new MySqlDataAdapter(cmdSelect);
                    DataTable dt = new DataTable();
                    adapter.Fill(dt); // Sonuçları DataTable'a doldurur.

                    dataGridView1.DataSource = dt; // Sonuçları dataGridView1 kontrolünde gösterir.
                }
                catch (Exception ex) // Bir hata oluşursa kullanıcıya hata mesajı gösterir.
                {
                    MessageBox.Show("Bir hata oluştu: " + ex.Message);
                }
            }
        }
    }
}
```



```

// button1 üzerine mouse ile gelindiğinde arka plan rengini açık maviye çevirir.
1 başvuru
private void button1_MouseEnter(object sender, EventArgs e)
{
    button1.BackColor = ControlPaint.Light(Color.RoyalBlue);
}

// button1 üzerinden mouse ayrıldığında arka plan rengini orijinaline döndürür.
1 başvuru
private void button1_MouseLeave(object sender, EventArgs e)
{
    button1.BackColor = Color.RoyalBlue;
}

// button1 tıklandığında çalışacak olan metod. Form3'ü açar ve mevcut formu (Form5) kapatır.
1 başvuru
private void button1_Click(object sender, EventArgs e)
{
    Form3 form3 = new Form3(); // Form3'ün bir örneğini oluşturur.
    form3.Show(); // Form3'ü gösterir.
    this.Close(); // Mevcut formu (Form5) kapatır.
}

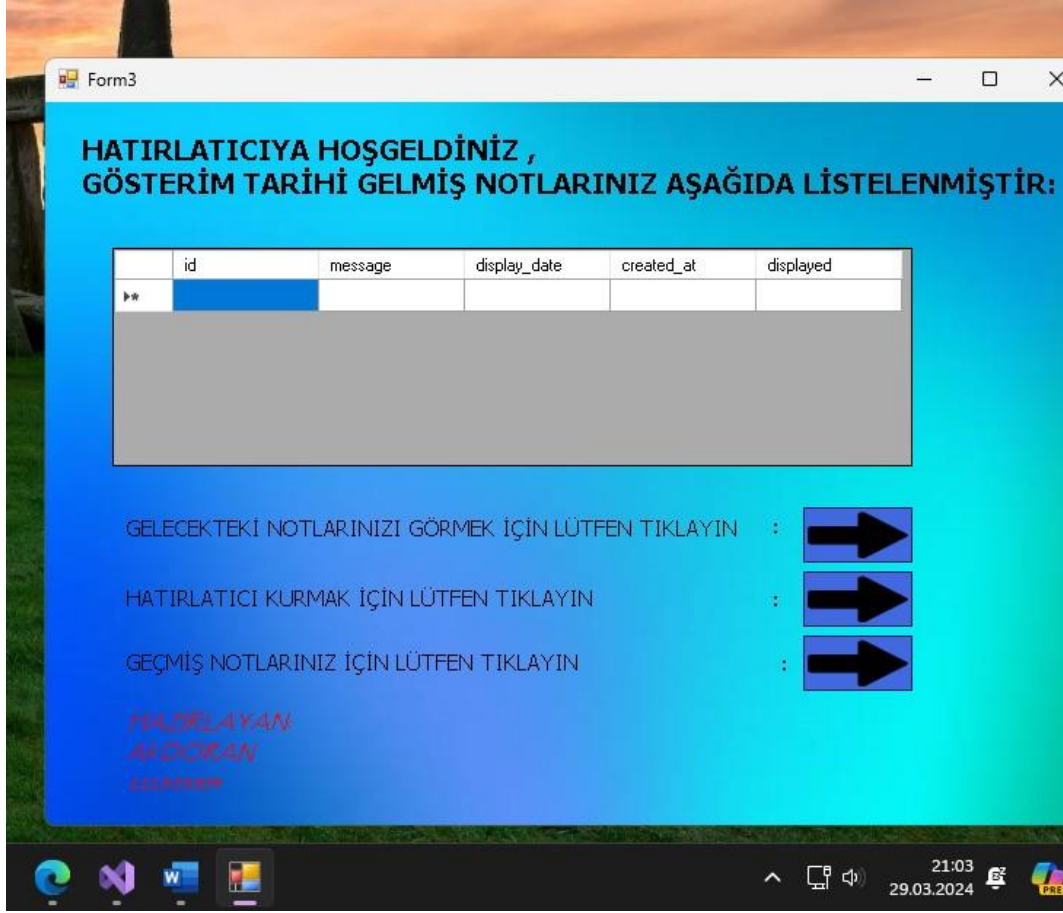
```

-UYGULAMA ÇALIŞTIRMA VE DENEME-

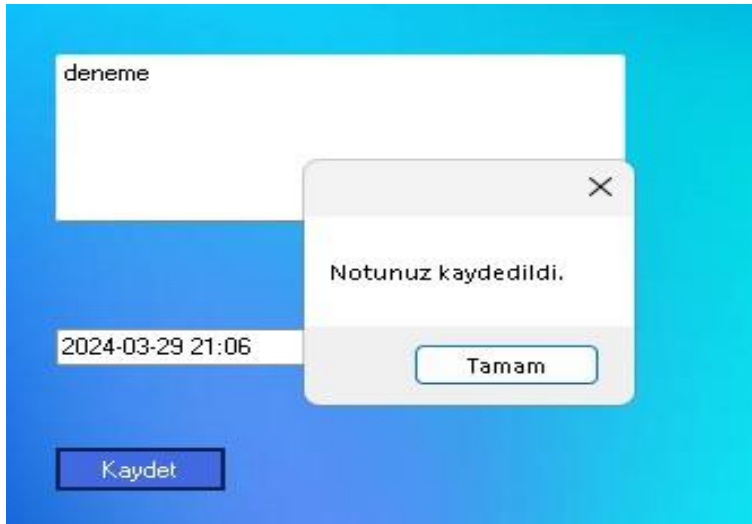
Uygulama çalışınca ana sayfaya girmeden önce bizi bir animasyonlu giriş karşılıyor .



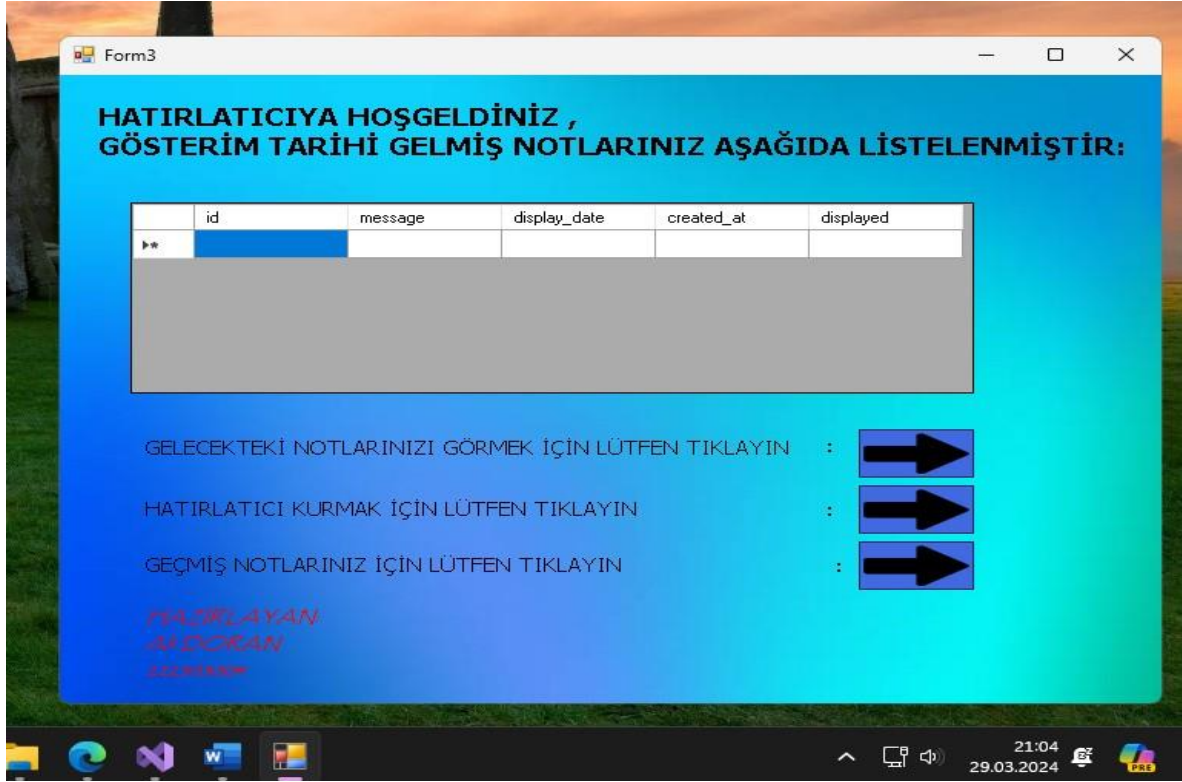
Daha sonra karşımıza ana sayfamız olan form3 çıkıyor. Burada zamanı gelen not veya zamanı geçmiş fakat uygulama açılmadığı için görülmemiş olan notlar listelenir. Burada 1 kere görüldükten sonra “displayed” sütunu 1 olacağından burdaki not listeden silinip geçmiş notların listelendiği form5 e gönderilir . Hemen bir deneme yapalım. Altta görüldüğü gibi saat 21.03 ve herhangi bir not gözüküyor. Öncelikle bir not oluşturmak için form2 ye gitmeliyiz bu yüzden ortadaki ok tuşuna basıyoruz.



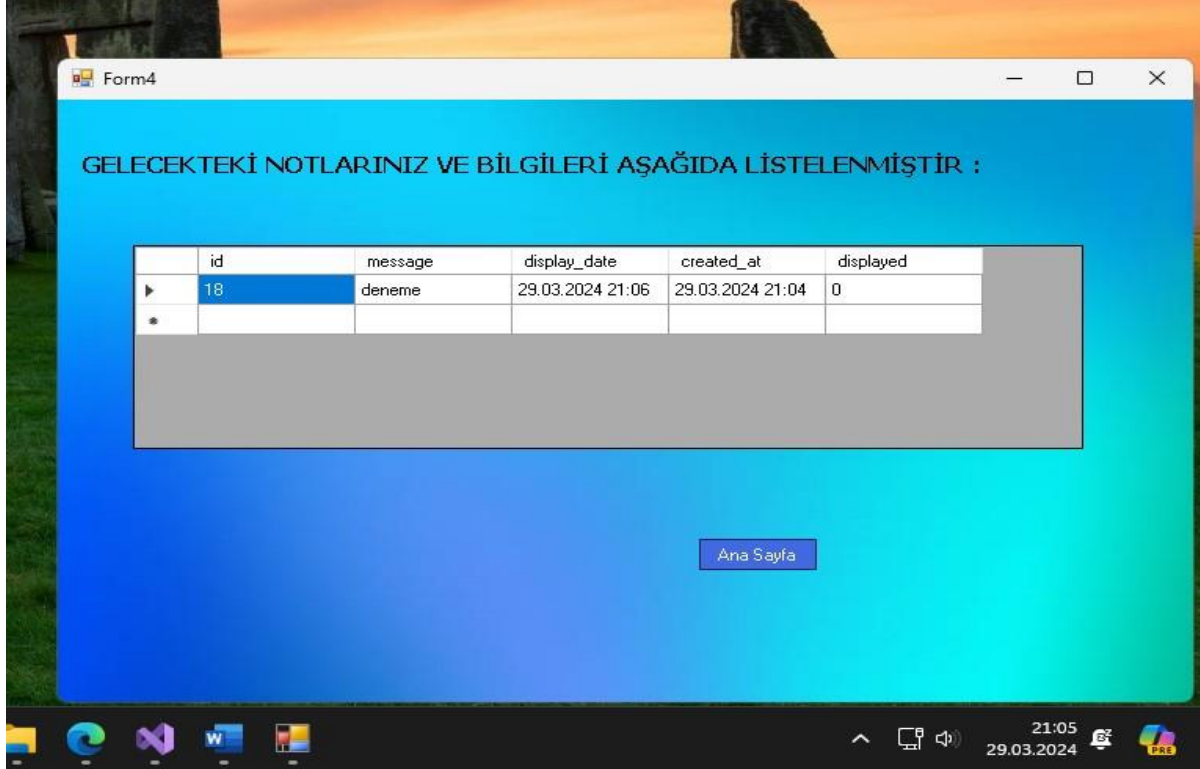
Form2 de “deneme” adında bir not ve zamanı 21.06 ya ayarlıyoruz.



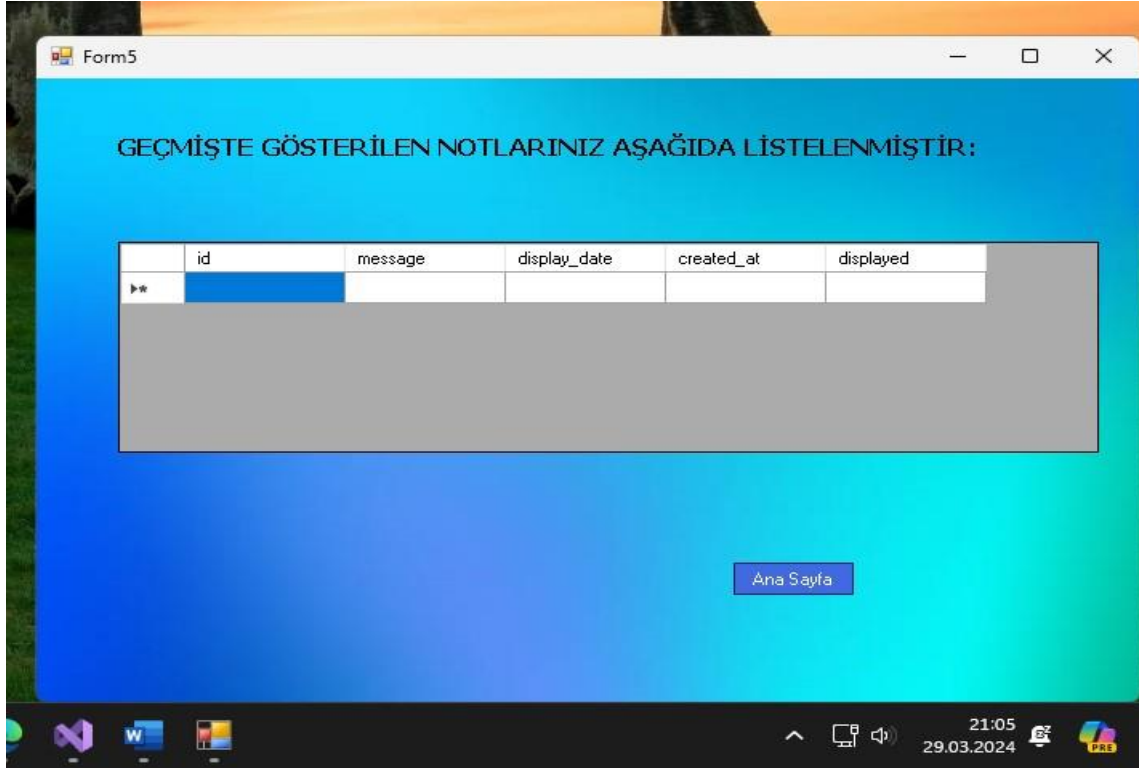
Görüldüğü gibi saat henüz 21.06 olmadığı için anasayfamızda herhangi bir not gözüküyor.



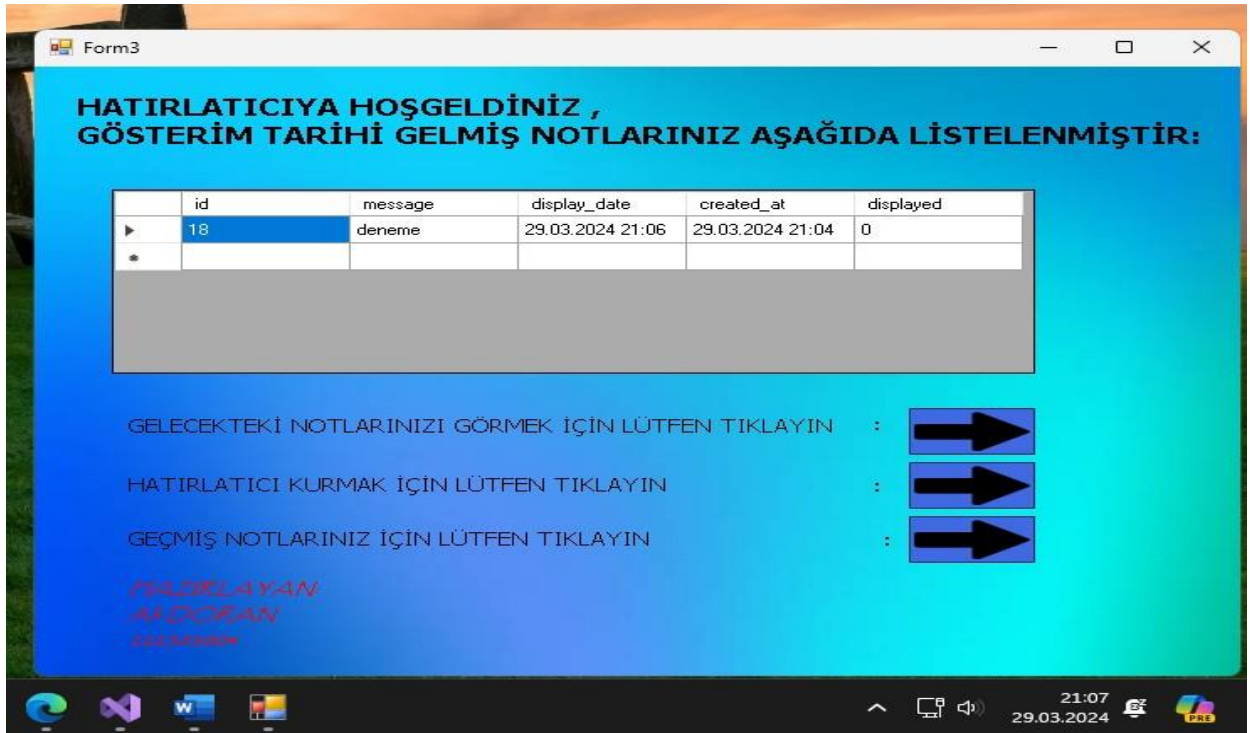
Görüldüğü gibi gelecekteki listemize baktığımızda notumuz burada . zamanı gelince not anasayfada gözüküp kendini burdaki listeden silecek.



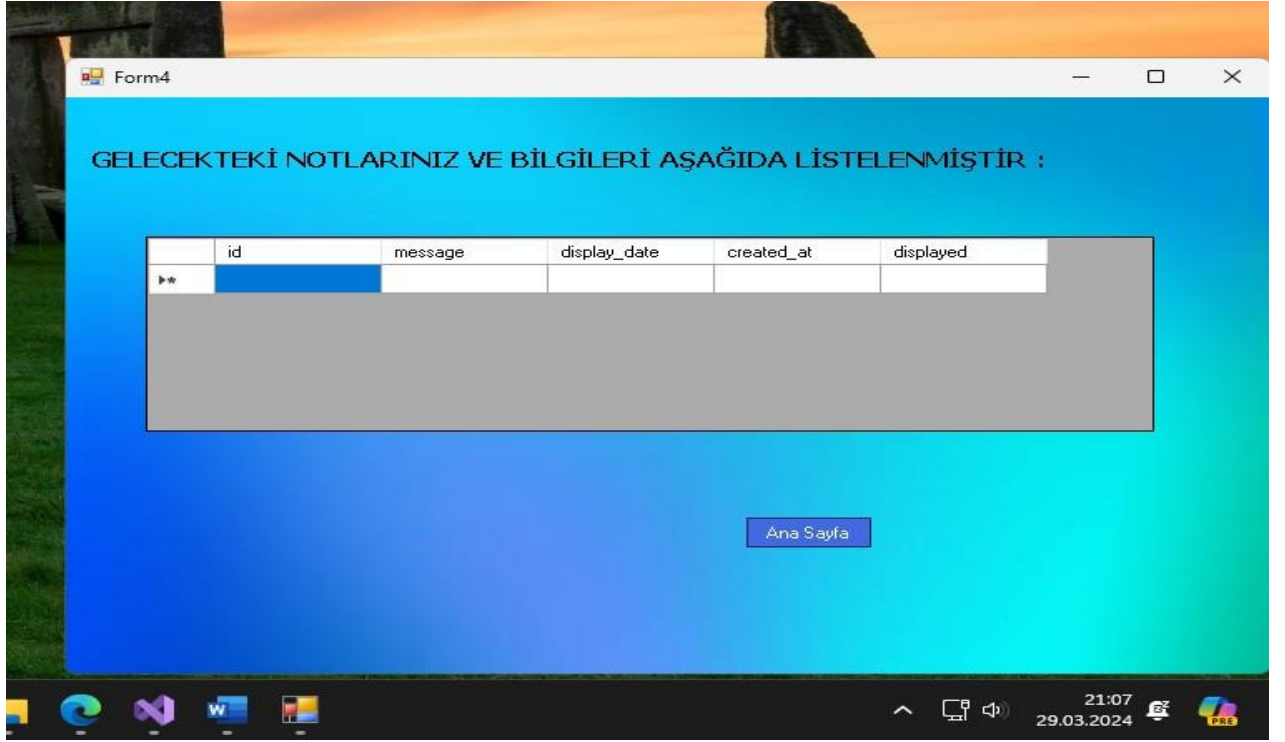
Geçmiş listemizde boş çünkü henüz bir notumuz listelenip görülmedi.



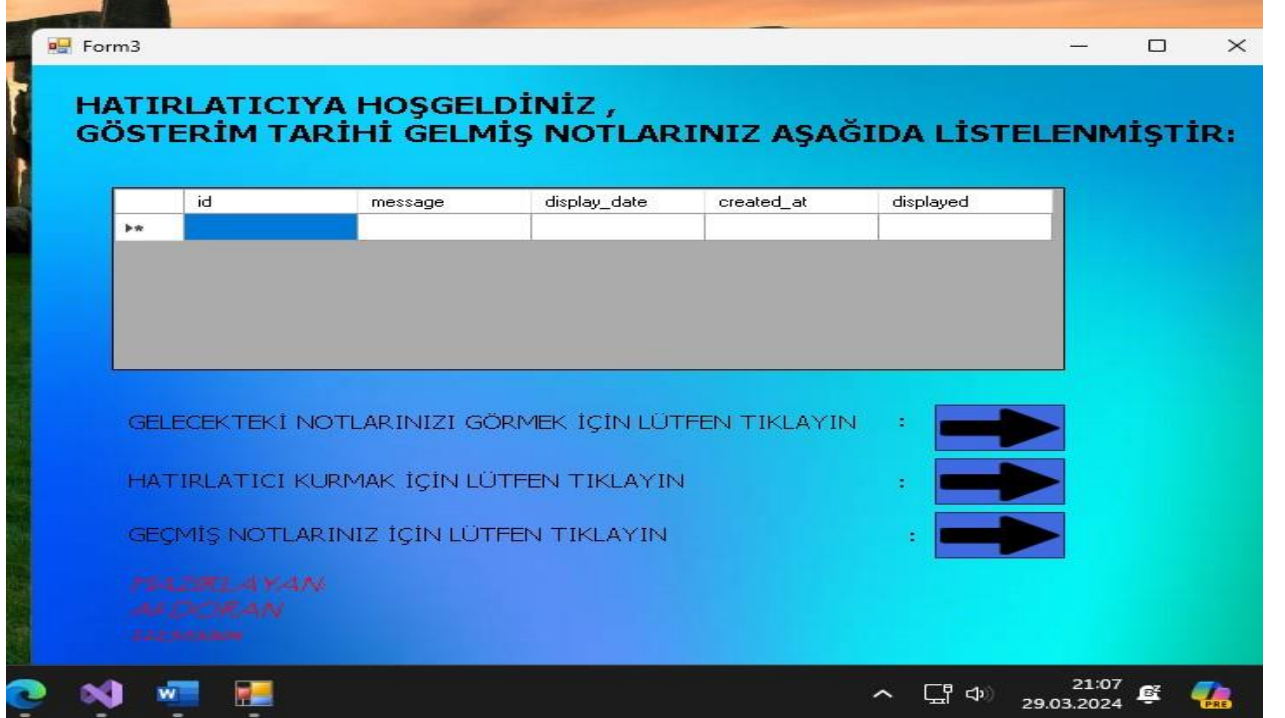
Görüldüğü gibi zaman gelince anasayfamızda listelendi ve biz bu notu gördük . artık başka bir sayfaya geçsek veya uygulamayı kapatıp yeniden başlatırsak notu öncden gördüğümüz için artık ana sayfada değil geçmiş notlarımızın listelendiği form5 te listelenecektir.



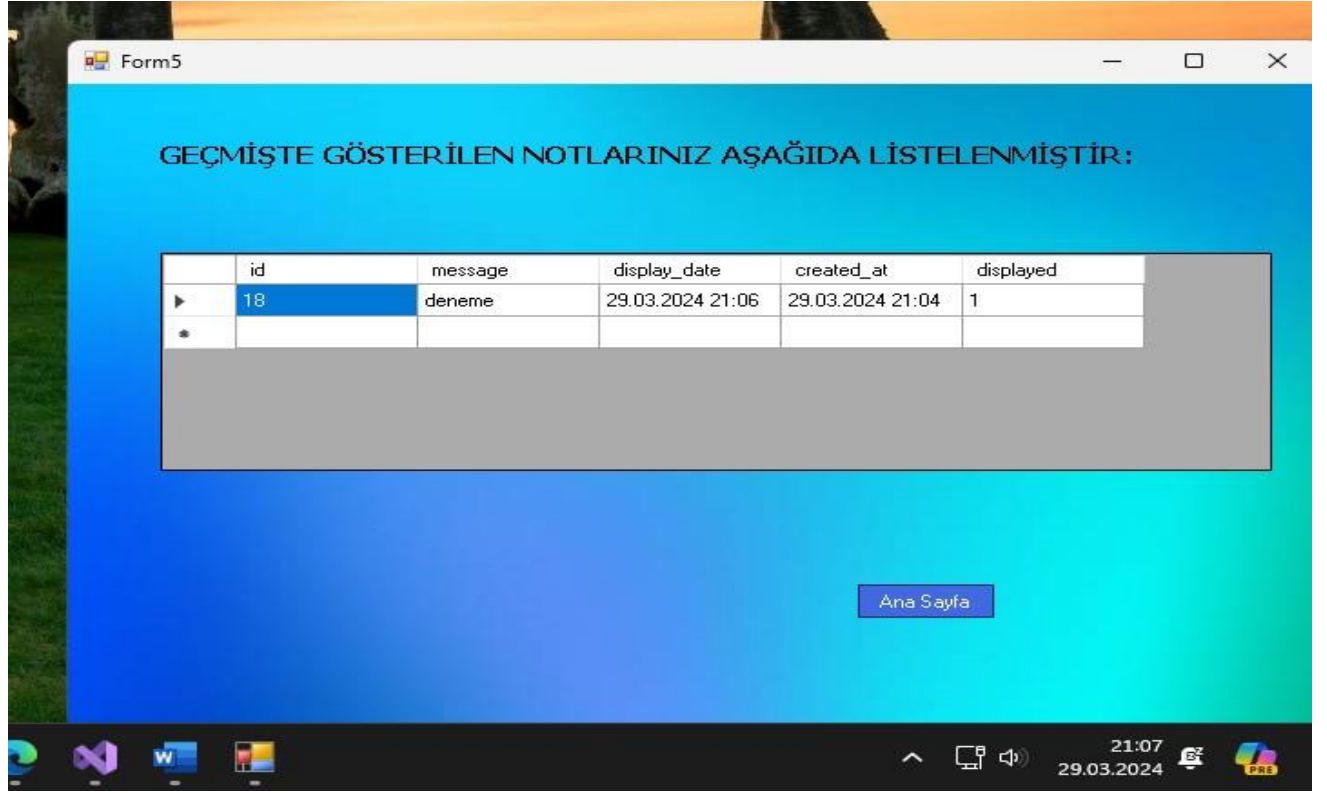
Gelecek not listesinden silindi çünkü artık zamanı geldi.



Ve gelecek listesinden anasayfaya geri döndüğümüzde not artık burada görülmeyecektir çünkü önceden gördüğümüz için “displayed=1” oldu ve burdaki listeden silinip geçmiş notların bulunduğu form5 e gönderildi.



Geçmiş notlara bakıldığında buraya gelmiş olduğunu görebiliyoruz.



Son olarak verinin sunucuya gidip gitmediğine bakabiliriz:

	id	message	display_date	created_at	displayed
<input type="checkbox"/> Düzenle Kopyala Sil	18	deneme	2024-03-29 21:06:59	2024-03-29 21:04:14	1
<input type="checkbox"/> Tümünü işaretle	Seçilileri: Düzenle Kopyala Sil Dışa aktar				
<input type="checkbox"/> Tümünü göster	Satır sayısı: 25	Satırları süz: Bu tabloda ara			

