

VIZUÁLIS PROGRAMOZÁS

GPTIFKN-VIZUPROG-1

ADMINISZTRÁTORI ADATBÁZIS KEZELŐ

DORMÁN LÁSZLÓ – RNH78K

BARANYAI ANDRÁS – AEN3WU



2022/23/1

NEUMANN JÁNOS EGYETEM – GAMF (FOSZK)

Bevezetés – Feladat / felület leírása

A projekt feladatunk egy adminisztrátori adatbáziskezelő program. A program elindításánál rögtön egy bejelentkezés fül fogad minket. A bejelentkezés ablak 2 mezőből áll, és egy bejelentkezés gombból. Csak akkor tudunk bejelentkezni, ha tudjuk a szükséges adatokat, máskülönben a program hibát fog kiírni, azt fogja mutatni, hogy rossz a felhasználónév, vagy a jelszó. Külön plusz dolog, hogy nem ad lehetőséget a felhasználónév kitalálására, mivel mind a kettő adat hibás lehet, így az illetéktelen felhasználó nem fogja tudni, hogy melyik adat is a hibás, szóval nem fog tudni belépni. Sikeres bejelentkezésnél egy újabb ablak fogad minket, ez az ablak az adatbázis szerkesztő ablak. A szerkesztő stílusos, dizájnban visszafogott, de igényes kinézetű. Az ablak tetején található az eszköztár, három darab „menu-item”, ez a fájl, a segítség, és az extrák. A fájlban belül kapunk plusz 3 elemet, ami a bezárás, mentés, és kijelentkezés. Ha a bezárásra megyünk, akkor a program teljesen bezáródik, és nincs lehetőségünk újra bejelentkezni, csak akkor hogyha újra elindítjuk a programot. Ha a mentésre megyünk, akkor elmentődik az adott munkamenet, de egyébként a program automatán elmenti a változtatásokat. Kijelentkezésnél végül pedig csak az adatbázis szerkesztőből lépünk ki, nem az egész programból, és kijelentkezésnél visszadob a bejelentkezés ablakhoz. A következő eszköz a segítség fül, ahol kapunk egy kezelési útmutatót, ami megmutatja nekünk hogy melyik elem mit csinál. Van egy „rólunk” fül, ahol egy kis információt kapunk a programot készítő cégről (ami jelen esetben mi vagyunk.) Végül pedig a kapcsolatfelvétel fül van hátra, ahol információt kapunk a cég elérhetőségéről, telefonszámról, emailcímről, és lakcímről. A következő eszköz az extrák fül, ahol meg lehet tekinteni a program névjegyét, verzióját. Van egy módosítás fül is, ahol a bejelentkezési adatokat tudjuk szerkeszteni, megváltoztatni, frissíteni. Ezen kívül lehetőségünk van témát váltani, ugyanis van sötét mód, világos mód, és statikus is. A szerkesztőben alapvetően 4 adatra van szükségünk: Név, Hozzáférés, Email, Biztonsági szint. Ezeknek a mezőknek a kitöltése kötelező, ha nem írunk be semmit, akkor nem fogjuk tudni rögzíteni az adatokat, ezáltal elmenteni sem. Közvetlen az adatok beírásától jobbra található maga az adatbázis, ami az adatokat mutatja, és listázza. Itt találjuk meg a rögzített adatokat, embereket. Az adatbázis alatt találunk 4 gombot: Rögzítés, Frissítés, Törlés, Reset. A rögzítés gomb adatokat rögzít a táblázatba, itt nem kell használnunk az ID-t, mivel automatán generálódik. A frissítés gomb módosít egy mezőt, itt kötelező megadni az ID-t, mivel így tudjuk biztosítani, hogy melyik adatot szeretnénk módosítani. A törlés gombnál csak az ID-t kell megadnunk mást nem, és ki tudjuk törölni a kívánt rekordot. A reset gomb a textboxba beírt

adatokat törli ki, így nem kell egyesével szenvedni az adatok kitörlésével. Van egy ID mezőnk a gomboktól balra, ahol betudjuk írni a kívánt ID-t, és a használata csak a kereséshez, törléshez, vagy frissítéshez kell. Itt megtalálható egy visszavonás gomb is, ami csak a keresést vonja vissza. A keresés gomb az ID megadása után működik. Az extráknál a bejelentkezések módosítása is ezen elvek alapján működnek (gombok, mezők stb.)

A program felépítése

LoginAblak.xaml.cs

```
public partial class LoginAblak : Window
{
    public LoginAblak()
    {
        InitializeComponent();
    }

    private void btnbekuldes_Click(object sender, RoutedEventArgs e)
    {
        SqlConnection sqlCon = new SqlConnection(@"Data Source=.\sqlexpress; Initial Catalog=LoginDB; Integrated Security=True;");
        try
        {
            if (sqlCon.State == ConnectionState.Closed) sqlCon.Open();
            String query = "SELECT COUNT(1) FROM tblUser WHERE Username=@Username AND Password=@Password";
            SqlCommand sqlCmd = new SqlCommand(query, sqlCon);
            sqlCmd.CommandType = CommandType.Text;
            sqlCmd.Parameters.AddWithValue("@Username", txtUsername.Text);
            sqlCmd.Parameters.AddWithValue("@Password", txtPassword.Password);
            int count = Convert.ToInt32(sqlCmd.ExecuteScalar());
            if (count == 1)
            {
                MainWindow dashboard = new MainWindow();
                dashboard.Show();
                this.Close();
            }
            else
            {
                MessageBox.Show("Rossz felhasználónév, vagy jelszó!", "Hibás!", MessageBoxButton.OK, MessageBoxImage.Error);
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
        finally
        {
            sqlCon.Close();
        }
    }
}
```

Ez a bejelentkezés gomb funkciója, csatlakozik az adatbázishoz, és összeveti a beírt adatokat az adatbáziséval, és ha nem egyeznek meg az adatok akkor a program hibát dob fel, vagyis felugrik egy messagebox hogy hibás adatok lettek megadva, viszont ha utána megadjuk a helyes adatokat a bejelentkezés sikeres lesz, és bezárja az ablakot a this.Close() -val, és megnyitja az adatbáziskezelő ablakot.

Adatok

```
if (sqlCon.State == ConnectionState.Closed) sqlCon.Open();
String query = "SELECT COUNT(1) FROM tblUser WHERE Username=@Username AND Password=@Password";
SqlCommand sqlCmd = new SqlCommand(query, sqlCon);
sqlCmd.CommandType = CommandType.Text;
sqlCmd.Parameters.AddWithValue("@Username", txtUsername.Text);
sqlCmd.Parameters.AddWithValue("@Password", txtPassword.Password);
```

If – else, ha sikerül akkor bezárja, ha nem akkor felugrik egy messagebox

```
if (count == 1)
{
    MainWindow dashboard = new MainWindow();
    dashboard.Show();
    this.Close();
}
else
{
    MessageBox.Show("Rossz felhasználónév, vagy jelszó!", "Hibás!", MessageBoxButton.OK, MessageBoxImage.Error);
}
```

MainWindow.xaml.cs

```
public partial class MainWindow : Window
{
    public MainWindow()
    {
        InitializeComponent();
        betoltesGrid();
    }

    SqlConnection con = new SqlConnection(@"Data Source=.\sqlexpress; Initial Catalog=LoginDB; Integrated Security=True;");

    public void reset()
    {
        nev_txt.Clear();
        hozzaferes_txt.Clear();
        email_txt.Clear();
        biztonsagi_txt.Clear();
        kereses_txt.Clear();
    }

    public void betoltesGrid()
    {
        SqlCommand cmd = new SqlCommand("select * from tblAdmin", con);
        DataTable dt = new DataTable();
        con.Open();
        SqlDataReader sdr = cmd.ExecuteReader();
        dt.Load(sdr);
        con.Close();
        datagrid.ItemsSource = dt.DefaultView;
    }

    private void resetbtn_Click(object sender, RoutedEventArgs e)
    {
        reset();
    }

    public bool haErvenyes()
    {
        if (nev_txt.Text == String.Empty)
        {
            MessageBox.Show("Név megadása kötelező!", "Sikertelen", MessageBoxButton.OK, MessageBoxImage.Error);
            return false;
        }

        if (hozzaferes_txt.Text == String.Empty)
        {
            MessageBox.Show("Hozzáférési jogosultság megadása kötelező!", "Sikertelen", MessageBoxButton.OK, MessageBoxImage.Error);
            return false;
        }

        if (email_txt.Text == String.Empty)
        {
            MessageBox.Show("Email megadása kötelező!", "Sikertelen", MessageBoxButton.OK, MessageBoxImage.Error);
            return false;
        }
    }
}
```

```

    if (biztonsagi_txt.Text == String.Empty)
    {
        MessageBox.Show("Biztonsági szint megadása kötelező!", "Sikertelen", MessageBoxButton.OK, MessageBoxImage.Error);
        return false;
    }
    return true;
}

public bool haErvenyes3()
{
    if (nev_txt.Text == String.Empty)
    {
        MessageBox.Show("Név megadása kötelező!", "Sikertelen", MessageBoxButton.OK, MessageBoxImage.Error);
        return false;
    }

    if (hozzaferes_txt.Text == String.Empty)
    {
        MessageBox.Show("Hozzáférési jogosultság megadása kötelező!", "Sikertelen", MessageBoxButton.OK, MessageBoxImage.Error);
        return false;
    }

    if (email_txt.Text == String.Empty)
    {
        MessageBox.Show("Email megadása kötelező!", "Sikertelen", MessageBoxButton.OK, MessageBoxImage.Error);
        return false;
    }

    if (biztonsagi_txt.Text == String.Empty)
    {
        MessageBox.Show("Biztonsági szint megadása kötelező!", "Sikertelen", MessageBoxButton.OK, MessageBoxImage.Error);
        return false;
    }

    if (kereses_txt.Text == String.Empty)
    {
        MessageBox.Show("ID megadása kötelező", "Sikertelen", MessageBoxButton.OK, MessageBoxImage.Error);
        return false;
    }

    return true;
}

public bool haErvenyes2()
{
    if (kereses_txt.Text == String.Empty)
    {
        MessageBox.Show("ID megadása kötelező", "Sikertelen", MessageBoxButton.OK, MessageBoxImage.Error);
        return false;
    }
    return true;
}

```

```

public bool haErvenyes2()
{
    if (kereses_txt.Text == String.Empty)
    {
        MessageBox.Show("ID megadása kötelező", "Sikertelen", MessageBoxButton.OK, MessageBoxImage.Error);
        return false;
    }
    return true;
}

private void rogzitesbtn_Click(object sender, RoutedEventArgs e)
{
    try
    {
        if (haErvenyes())
        {
            SqlCommand cmd = new SqlCommand("INSERT INTO tblAdmin VALUES (@Name, @Acces, @Email, @PriorityLvl)", con);
            cmd.CommandType = CommandType.Text;
            cmd.Parameters.AddWithValue("@Name", nev_txt.Text);
            cmd.Parameters.AddWithValue("@Acces", hozzaferes_txt.Text);
            cmd.Parameters.AddWithValue("@Email", email_txt.Text);
            cmd.Parameters.AddWithValue("@PriorityLvl", biztonsagi_txt.Text);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            betoltesGrid();
            MessageBox.Show("Sikeres rögzítés", "Mentve", MessageBoxButton.OK, MessageBoxImage.Information);
            reset();
        }
    }
    catch (SqlException ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

```

private void torlesbtn_Click(object sender, RoutedEventArgs e)
{
    con.Open();
    SqlCommand cmd = new SqlCommand("delete from tblAdmin WHERE ID = " + kereses_txt.Text + " ", con);
    try
    {
        cmd.ExecuteNonQuery();
        MessageBox.Show("A Record törölve lett.", "Törölve!", MessageBoxButton.OK, MessageBoxImage.Information);
        con.Close();
        reset();
        betoltesGrid();
        con.Close();
    }
    catch (SQLException ex)
    {
        MessageBox.Show("Sikertelen törlés. ID megadása hibás. Hiba! " + ex.Message);
    }
    finally
    {
        con.Close();
    }
}

private void frissitesbtn_Click(object sender, RoutedEventArgs e)
{
    con.Open();
    SqlCommand cmd = new SqlCommand("update tblAdmin set Name = '" + nev_txt.Text + "', Accos = '" + hozzaferes_txt.Text + "', Email = '" + email_txt.Text + "', PriorityLvl = '" + biztonsagi_txt.Text + "' WHERE ID = '" + kereses_txt.Text + "' ", con);
    try
    {
        if (haErvenyes3())
        {
            cmd.ExecuteNonQuery();
            MessageBox.Show("A Record sikeresen frissítve lett!", "Frissítve!", MessageBoxButton.OK, MessageBoxImage.Information);
            reset();
        }
    }
    catch (SQLException ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        con.Close();
        betoltesGrid();
    }
}

private void bozarasmenu_Click(object sender, RoutedEventArgs e)
{
    this.Close();
}

```

```

private void mentesmenu_Click(object sender, RoutedEventArgs e)
{
    MessageBox.Show("A program automatikusan menti magát! :) De a mentés sikeres.", "Mentve!", MessageBoxButton.OK, MessageBoxImage.Information);
}

private void segitsagmenu_Click(object sender, RoutedEventArgs e)
{
    Segitsag megnyitas = new Segitsag();
    megnyitas.Show();
}

private void rolukenmu_Click(object sender, RoutedEventArgs e)
{
    MessageBox.Show("Mi egy kis cég vagyunk, ami kisebb programok megírásával foglalkozik. \n A cég 2022-ben alapított meg Herberts Company néven. \n Bővebb információt weboldalunkon olvashat.", "Rólunk", MessageBoxButton.OK, MessageBoxImage.Information);
}

private void kapcsolatmenu_Click(object sender, RoutedEventArgs e)
{
    MessageBox.Show("Vádd fel volnék a kapcsolatot akár személyesen akár online: \n Tel.: 06 1 999 999 \n Email: dovfejlesztes@info.hu \n Helyszín: 2151 Fét Vörösmarty Utca 1", "Kapcsolat felvétel", MessageBoxButton.OK, MessageBoxImage.Information);
}

private void bejelentkezesmenu_Click(object sender, RoutedEventArgs e)
{
    LoginEdit megnyitas2 = new LoginEdit();
    megnyitas2.Show();
}

private void kijelentkezesmenu_Click(object sender, RoutedEventArgs e)
{
    LoginBalk megnyitas3 = new LoginBalk();
    megnyitas3.Show();
    this.Close();
}

private void nevjegymenu_Click(object sender, RoutedEventArgs e)
{
    MessageBox.Show("Név: Adatházkezelő alkalmazás \n Verzió szm: V1.0.0 OpenBeta", "Alkalmazás névjegye", MessageBoxButton.OK, MessageBoxImage.Information);
}

```

```

private void keresesgomb_Click(object sender, RoutedEventArgs e)
{
    try
    {
        if (haErvenyes2())
        {
            SqlCommand command = new SqlCommand("select * from tblAdmin where ID = " + int.Parse(kereses_txt.Text) + " ", con);
            SqlDataAdapter sd = new SqlDataAdapter(command);
            DataTable dt = new DataTable();
            sd.Fill(dt);
            datagrid.ItemsSource = dt.DefaultView;
            reset();
        }
    }
    catch (SQLException ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void visszabtn22_Click(object sender, RoutedEventArgs e)
{
    con.Open();
    con.Close();
    betoltesGrid();
    reset();
}

```

```

private void vilagostema_click(object sender, RoutedEventArgs e)
{
    nev_txt.Background = Brushes.White;
    nev_txt.Foreground = Brushes.Black;
    nev_txt.BorderBrush = Brushes.Black;

    hozzaferes_txt.Background = Brushes.White;
    hozzaferes_txt.Foreground = Brushes.Black;
    hozzaferes_txt.BorderBrush = Brushes.Black;

    kereses_txt.Background = Brushes.White;
    kereses_txt.Foreground = Brushes.Black;
    kereses_txt.BorderBrush = Brushes.Black;

    email_txt.Background = Brushes.White;
    email_txt.Foreground = Brushes.Black;
    email_txt.BorderBrush = Brushes.Black;

    biztonsagi_txt.Background = Brushes.White;
    biztonsagi_txt.Foreground = Brushes.Black;
    biztonsagi_txt.BorderBrush = Brushes.Black;

    gridalap.Background = Brushes.White;

    datagrid.Background = Brushes.White;
    datagrid.Foreground = Brushes.Black;
    datagrid.BorderBrush = Brushes.Black;
    datagrid.RowBackground = Brushes.White;
    datagrid.HorizontalGridLinesBrush = Brushes.Black;
    datagrid.VerticalGridLinesBrush = Brushes.Black;

    admintext.Foreground = Brushes.Black;
    nevtext.Foreground = Brushes.Black;
    emailtext.Foreground = Brushes.Black;
    hferestext.Foreground = Brushes.Black;
    biztonsagitext.Foreground = Brushes.Black;
    idtext.Foreground = Brushes.Black;
}

```

Programkód leírás

betoltesGrid(); //Grid betöltése miatt van rá szükség, ez fogja nekünk betölteni a táblázatot a Gridbe.

SqlConnection con = new SqlConnection(@"Data Source=.\sqlexpress; Initial Catalog=LoginDB; Integrated Security=True;"); //Csatlakoztatja az adatbázist a programhoz.

public void reset() //Ez egy olyan funckió ami a textboxokat egybe foglalja és ha van bele valami írva törli.

```
{
    nev_txt.Clear();
    hozzaferes_txt.Clear();
    email_txt.Clear();
    biztonsagi_txt.Clear();
    kereses_txt.Clear();
}
```

public void betoltesGrid() //Ez jeleníti meg a gridben az adatbázis adatait.

```
{
    SqlCommand cmd = new SqlCommand("select * from tblAdmin", con);
    DataTable dt = new DataTable();
    con.Open();
    SqlDataReader sdr = cmd.ExecuteReader();
    dt.Load(sdr);
    con.Close();
    datagrid.ItemsSource = dt.DefaultView;
}
```

private void resetbtn_Click(object sender, RoutedEventArgs e) //ez egy gombnak a funkciója ha lenyomják akkor végve megy a reset2 dolga, vagyis kitörli a textboxokból az információkat.

```
{
    reset();
}
```

public bool haErvenyes() //ez egy ellenőrző, azt figyeli, hogy van e írva valami a textboxba vagy nincs, és ha nincs feldob egy messageboxot, hogy az a mező nincsen kitöltve.

public bool haErvenyes3() //ez szintén egy ellenőrző ami azt figyeli, hogy van e írva valami a textboxba vagy nincs, és ha nincs feldob egy messageboxot, hogy az a mező nincsen kitöltve.

public bool haErvenyes2() //és ez szintén egy ellenőrző ami azt figyeli, hogy van e írva valami a textboxba vagy nincs, és ha nincs feldob egy messageboxot, hogy az a mező nincsen kitöltve.

private void rogzitesbtn_Click(object sender, RoutedEventArgs e) //ez a rögzítés gomb funkciója ha lenyomják végbe megy az adatok rögzítése. úgy működik hogy,

adattal kapcsolatosan //a textboxba írt adatokat összekapcsolja az
adatbázissal és úgy a megadott helyre menti a táblázatban,

ellenőrzés //közben figyeli, hogy minden mező kivan e
töltve, ugye ez a haErvenyes() rész feladata,

adattal kapcsolatosan //ha van benne valami engedélyezi a rögzítést,
illetve a bevitel ellenőrzött,

adattal kapcsolatosan //ha van valami hiba a rögzítéssel kapcsolatban
azt kiírja.

`private void torlesbtn_Click(object sender, RoutedEventArgs e) //Ez a törlés gomb funkciója`
ha megnyomják ugye kitöröl egy recordot az adatbázis táblázatából,
//a törlés ellenőrzött ha nincs ID megadva vissza
dob egy hibát és nem hajt végre semmit ha van megadva ID akkor töröl.

`private void frissitesbtn_Click(object sender, RoutedEventArgs e) //Ez a frissítés gomb`
funkciója ha megnyomjuk akkor a benne lévő kódsor lefut,
//ez is ellenőrzött tehát ha a megfelelő mezők
nincsenek megadva akkor a frissítés nem végződik el és hibát dob vissza nekünk,
//egyrészt ez lehet olyan hiba amit a
haervenyes2() dobhat vissza hogy nincs kitöltve egy textbox,
//vagy adatbázis hibát dobhat vissza amit kiír
hogyan mi a hiba.

`private void bezarasmenu_click(object sender, RoutedEventArgs e) //a toolbaron belül a`
bezárásra kattintva bezárja az egész alkalmazást és kilép belőle a `this.close()` segítségével

`private void mentesmenu_click(object sender, RoutedEventArgs e) //a toolbaron belül a`
mentésre kattintva kiad egy messageboxot hogy a mentés sikeres és hogy automatikusan ment
amúgy is a rendszer.

`private void segitsegmenu_click(object sender, RoutedEventArgs e) //a toolbaron belül a`
segítségre kattintva egy ablak ugrik fel amin egy képen belül elmagyarázza hogy kell kezelni
az alkalmazást

`private void rolunkmenu_click(object sender, RoutedEventArgs e) //a toolbaron belül a`
rolunkra kattintva felugrik egy információs messagebox ami leír rólunk 1-2 dolgot.

`private void kapcsolatmenu_click(object sender, RoutedEventArgs e) // a toolbaron belül a`
kapcsolatra kattintva egy message box ugrik fel ahol a kapcsolat felvételről találunk egy két
dolgot

`private void bejelentkezesmenu_click(object sender, RoutedEventArgs e) //a toolbaron belül a`
módosításokra kattintva behoz egy ablakot ahol a belépsi adatokat lehet módosítani.

`private void kijelentkezesmenu_click(object sender, RoutedEventArgs e) //a toolbaron belül a`
kijelentkezésre kattintva ez kijelentkeztet az alkalmazásból és vissza visz a bejelentkezési
felületre.

`private void nevjegymenu_click(object sender, RoutedEventArgs e) // a toolbaron belül a`
nevjegyre kattintva behozza az alkalmazás verzió számát egy messagebox segítségével.

`private void keresesgomb_Click(object sender, RoutedEventArgs e) //ez a keresés funkció`
megvalósítása ID alapján lehet keresgélni az adatbázis táblázatában a könnyebb megtalálás
érdekében.

`private void visszabtn22_Click(object sender, RoutedEventArgs e) //mivel a keresés`
funkcióban az adatbázis a keresett ID-re ugrik és csak azt mutatja ezért kell egy vissza gomb is
ami újra kilistázza az összes ID-t

`private void vilagostema_click(object sender, RoutedEventArgs e) //A textboxok, textek, datagrid, gridek átszínezése, ez valósítja meg a TÉMA funkció működését.`

`private void sotettema_click(object sender, RoutedEventArgs e) //A textboxok, textek, datagrid, gridek átszínezése, ez valósítja meg a TÉMA funkció működését.`

`private void kektema_click(object sender, RoutedEventArgs e) //A textboxok, textek, datagrid, gridek átszínezése, ez valósítja meg a TÉMA funkció működését.`

Segítség.xaml.cs

```
public partial class Segitseg : Window
{
    public Segitseg()
    {
        InitializeComponent();
    }

    private void bezarasbtn_Click(object sender, RoutedEventArgs e)
    {
        this.Close();
    }
}
```

`private void bezarasbtn_Click(object sender, RoutedEventArgs e) //EZ egy gomb funkciója ha rákattint akkor bezárja ami a this.close hajt végre.`

LoginEdit.xaml.cs

```
public partial class LoginEdit : Window
{
    public LoginEdit()
    {
        InitializeComponent();
        BetoltGrid2();
    }

    SqlConnection con = new SqlConnection(@"Data Source=.\sqlexpress; Initial Catalog=LoginDB; Integrated Security=True;");

    public void reset2()
    {
        felhasznalonev_txt.Clear();
        jelszo_txt.Clear();
        kereses2_txt.Clear();
    }

    public void BetoltGrid2()
    {
        SqlCommand cmd = new SqlCommand("select * from tblUser", con);
        DataTable dt = new DataTable();
        con.Open();
        SqlDataReader sdr = cmd.ExecuteReader();
        dt.Load(sdr);
        con.Close();
        datagrid2.ItemsSource = dt.DefaultView;
    }

    private void resetbtn2_Click(object sender, RoutedEventArgs e)
    {
        reset2();
    }

    public bool haErvenyes()
    {
        if (felhasznalonev_txt.Text == String.Empty)
        {
            MessageBox.Show("Felhasználónév megadása kötelező!", "Sikertelen", MessageBoxButton.OK, MessageBoxImage.Error);
            return false;
        }

        if (jelszo_txt.Text == String.Empty)
        {
            MessageBox.Show("Jelszó megadása kötelező!", "Sikertelen", MessageBoxButton.OK, MessageBoxImage.Error);
            return false;
        }

        return true;
    }
}
```

```

public bool haErvenyes2()
{
    if (felhasznalonev_txt.Text == String.Empty)
    {
        MessageBox.Show("Felhasználónév megadása kötelező!", "Sikertelen", MessageBoxButton.OK, MessageBoxImage.Error);
        return false;
    }

    if (jelszo_txt.Text == String.Empty)
    {
        MessageBox.Show("Jelszó megadása kötelező!", "Sikertelen", MessageBoxButton.OK, MessageBoxImage.Error);
        return false;
    }

    if (kereses2_txt.Text == String.Empty)
    {
        MessageBox.Show("ID megadása kötelező!", "Sikertelen", MessageBoxButton.OK, MessageBoxImage.Error);
        return false;
    }

    return true;
}

private void rogztitesbtn2_Click(object sender, RoutedEventArgs e)
{
    try
    {
        if (haErvenyes())
        {
            SqlCommand cmd = new SqlCommand("INSERT INTO tblUser VALUES (@UserName, @Password)", con);
            cmd.CommandType = CommandType.Text;
            cmd.Parameters.AddWithValue("@UserName", felhasznalonev_txt.Text);
            cmd.Parameters.AddWithValue("@Password", jelszo_txt.Text);
            con.Open();
            cmd.ExecuteNonQuery();
            con.Close();
            BetoltGrid2();
            MessageBox.Show("Sikeres rögzítés", "Mentve", MessageBoxButton.OK, MessageBoxImage.Information);
            reset2();
        }
    }
    catch (SQLException ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

```

private void torlesbtn2_Click(object sender, RoutedEventArgs e)
{
    con.Open();
    SqlCommand cmd = new SqlCommand("delete from tblUser WHERE UserId = "+kereses2_txt.Text+" ", con);
    try
    {
        cmd.ExecuteNonQuery();
        MessageBox.Show("A Record törölve lett.", "Törölve!", MessageBoxButton.OK, MessageBoxImage.Information);
        con.Close();
        reset2();
        BetoltGrid2();
        con.Close();
    }
    catch (SQLException ex)
    {
        MessageBox.Show("Sikertelen törlés. Hiba! " + ex.Message);
    }
    finally
    {
        con.Close();
    }
}

private void frissitesbtn2_Click(object sender, RoutedEventArgs e)
{
    con.Open();
    SqlCommand cmd = new SqlCommand("update tblUser set UserName = '"+felhasznalonev_txt.Text+"', Password = '"+jelszo_txt.Text+" ' WHERE UserId = '"+kereses2_txt.Text+" ' ", con);
    try
    {
        if (haErvenyes2())
        {
            cmd.ExecuteNonQuery();
            MessageBox.Show("A Record sikeresen frissítve lett!", "Frissítve!", MessageBoxButton.OK, MessageBoxImage.Information);
            reset2();
        }
    }
    catch (SQLException ex)
    {
        MessageBox.Show(ex.Message);
    }
    finally
    {
        con.Close();
        BetoltGrid2();
    }
}

```

```

private void bezarasbtn2_Click(object sender, RoutedEventArgs e)
{
    this.Close();
}

```

BetoltGrid2(); //Grid betöltése miatt van rá szükség, ez fogja nekünk betölteni a táblázatot a Gridbe.

```
SqlConnection con = new SqlConnection(@"Data Source=.\sqlexpress; Initial Catalog=LoginDB; Integrated Security=True;"); //Csatlakoztatja az adatbázist a programhoz.
```

```
public void reset2() //Ez egy olyan funkció ami a textboxokat egybe foglalja és ha van bele valami írva törli.
```

```
public void BetoltGrid2() //Ez jeleníti meg a gridben az adatbázis adatait.
```

```
private void resetbtn2_Click(object sender, RoutedEventArgs e) //ez egy gombnak a funkciója ha lenyomják akkor végve megy a reset2 dolga, vagyis kitörli a textboxokból az információkat.
```

```
public bool haErvenyes() //ez egy ellenőrző, azt figyel, hogy van e írva valami a textboxba vagy nincs, és ha nincs feldob egy messageboxot, hogy az a mező nincsen kitöltve.
```

```
public bool haErvenyes2() //ez szintén egy ellenőrző ami azt figyel, hogy van e írva valami a textboxba vagy nincs, és ha nincs feldob egy messageboxot, hogy az a mező nincsen kitöltve.
```

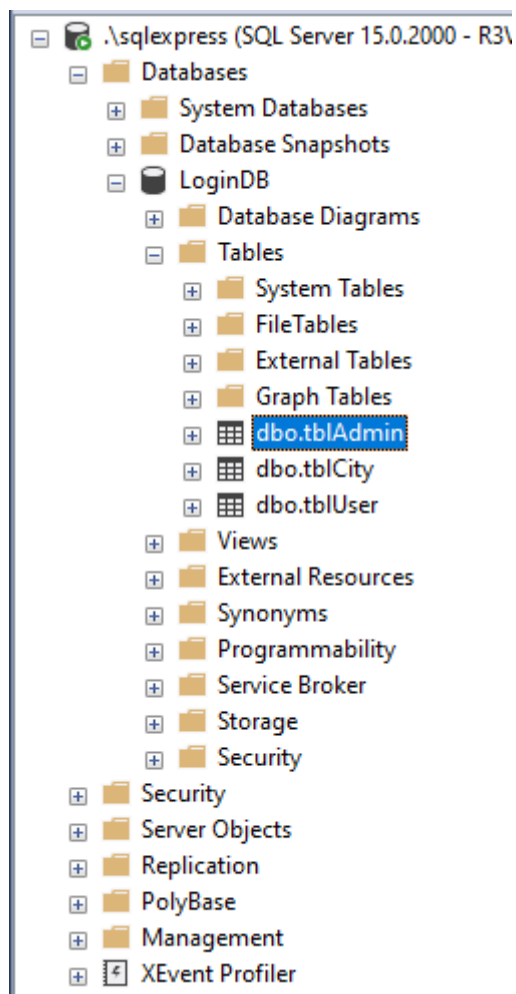
```
private void rogzitesbtn2_Click(object sender, RoutedEventArgs e) //ez a rögzítés gomb funkciója ha lenyomják végbe megy az adatok rögzítése. úgy működik hogy,  
//a textboxba írt adatokat összekapcsolja az  
adatbázissal és úgy a megadott helyre menti a táblázatban,  
//közben figyel, hogy minden mező kivan e  
töltve, ugye ez a haErvenyes() rész feladata,  
//ha van benne valami engedélyezi a rögzítést,  
illetve a bevitel ellenőrzött,  
//ha van valami hiba a rögzítéssel kapcsolatban  
azt kiírja.
```

```
private void torlesbtn2_Click(object sender, RoutedEventArgs e) //Ez a törlés gomb funkciója ha megnyomják ugye kitöröl egy recordot az adatbázis táblázatából,  
//a törlés ellenőrzött ha nincs ID megadva vissza  
dob egy hibát és nem hajt végre semmit ha van megadva ID akkor töröl.
```

```
private void frissitesbtn2_Click(object sender, RoutedEventArgs e) //Ez a frissítés gomb funkciója ha megynomjuk akkor a benne lévő kódsor lefut,  
//ez is ellenőrzött tehát ha a megfelelő mezők  
nincsenek megadva akkor a frissítés nem végződik el és hibát dob vissza nekünk,  
//egyrészt ez lehet olyan hiba amit a  
haErvenyes2() dobhat vissza hogy nincs kitöltve egy textbox,  
//vagy adatbázis hibát dobhat vissza amit kiír  
hogyan mi a hiba.
```

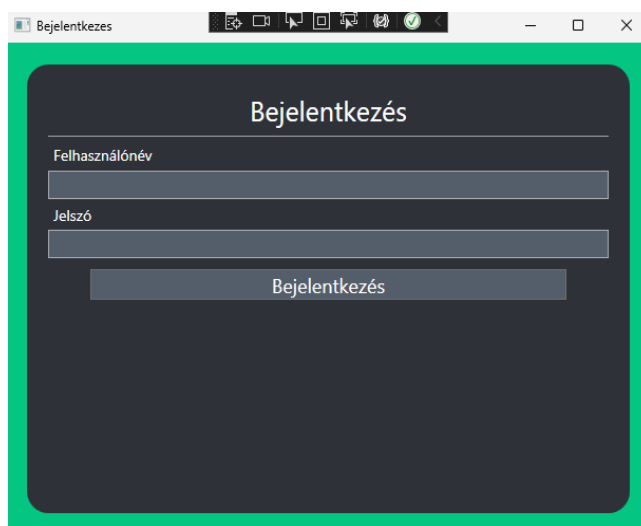
```
private void bezarasbtn2_Click(object sender, RoutedEventArgs e) //Ez a bezárás gomb funkciója ha lenyomják akkor bezárja az ablakot ugye amit a this.Close() hajt végre
```

Adatbázis



Az adatbázis Microsoft SQL Server Management Studio 18-al érhető el, kezelhető. Látható a 3db tábla, ami a tblAdmin, tblCity, és tblUser-ra hallgat. Az adatbázis neve LoginDB.

Alkalmazás kinézet



Adatbázis szerkesztő

Fájl Segítség Extrák

Adminisztrátori adatbázis szerkesztő

Név:

Hozzáférés:

Email:

Biztonsági szint:

ID	Name	Access	Email
1	Dormán László	Tulajdonos	dorman.laszlo@gmail.com
2	Baranyai András	Tulajdonos	bandras@gmail.com
3	Lukácsi Dániel	Fejlesztő	lukacsidani@gmail.com
4	Szabó Adrián	Fejlesztő	szaboadrian@gmail.com

ID:

Rögzítés

Frissítés

Törlés

Reset

Keresés

Visszavonás

Adatbázis szerkesztő

Fájl Segítség Extrák

Bejelentkezési adatok módosítása

Név:

Hozzáférés:

Email:

Biztonsági szint:

Felhasználónév:

Jelszó:

ID:

UserId	UserName	Password
1	admin	admin123

Bezárás

Rögzítés

Frissítés

Törlés

Reset

Fájl Segítség Extrák

Módosítások

Névjegy

Témák

Az alkalmazás témájának módosítása.

Világos

Sötét

Statikus

Név:

Fájl Segítség Extrák

Bezárás

Mentés

Kijelentkezés

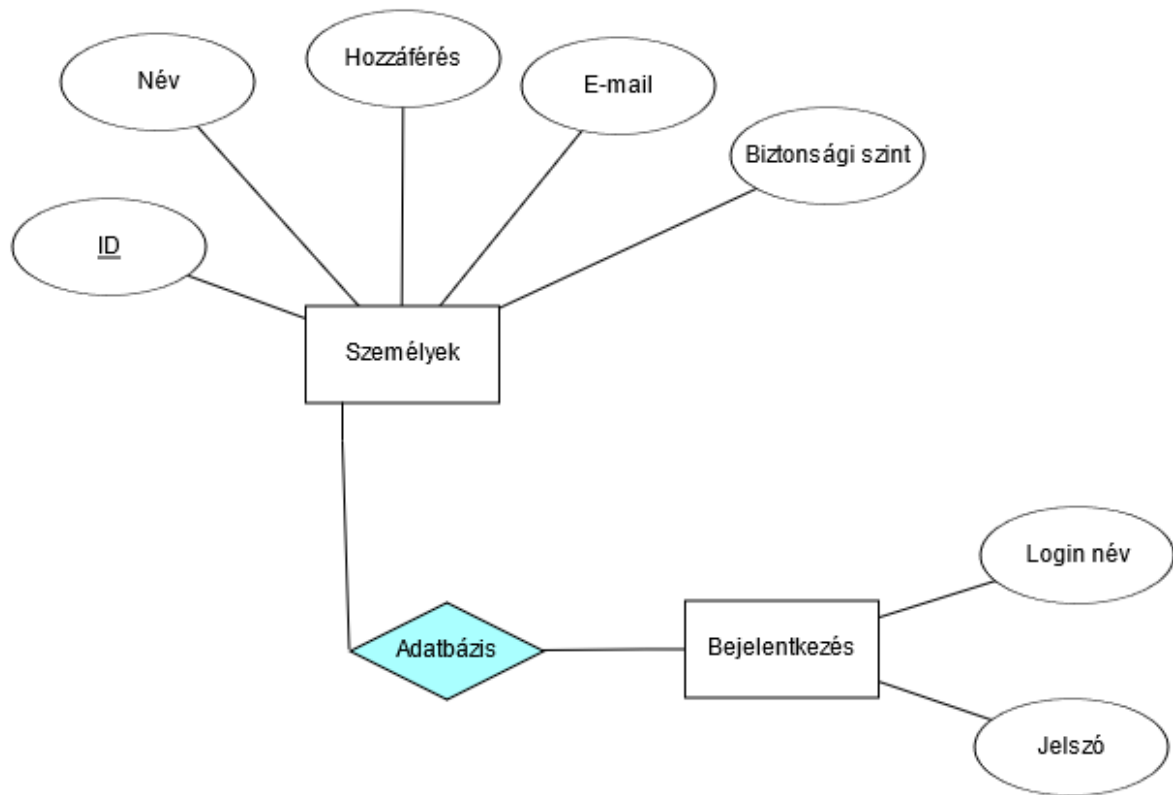
Fájl Segítség Extrák

Kezelési útmutató

Rólunk

Kapcsolat felvétel

EK diagramm



Osztály diagramm

