

Dunaújvárosi Egyetem Bánki Donát Technikum

Projekt feladat dokumentáció

Projekt tervező: Radnai Szabolcs

Projekt címe: Adatbázis-kezelő

Osztály: 13.C/ IpInf

Feladat meghatározás:

A feladat az volt, hogy készítsek egy Windows Forms alkalmazást, amely egy MySQL adatbázissal kommunikál, és lehetővé teszi az adatok hozzáadását, törlését, frissítését és megjelenítését egy DataGridView vezérlőn keresztül. Az alábbiakban részletesen elmagyarázom, hogy mi történik lépésről lépésre:

Osztály és Konstruktor:

```
public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
        gridview();
    }
}
```

A Form1 osztály a Windows Forms alkalmazás fő űrlapja. A konstruktorban meghívódik az InitializeComponent() metódus, amely inicializálja a formot és annak vezérlőit (pl. gombok, szövegdobozok). Az gridView() metódus is meghívódik, amely betölti az adatokat az adatbázisból és megjeleníti őket a DataGridView vezérlőben.

Adatok hozzáadása az adatbázishoz (button1 Click):

```
private void button1_Click(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection("server=localhost;user
id=root;database=projectmysql");
    con.Open();
    MySqlCommand cmd = new MySqlCommand("INSERT INTO project
(`name`, `address`, `phone`, `email`) VALUES ('" + Nametxt.Text +
"', '" + Addtxt.Text + "', '" + Phonetxt.Text + "', '" + emailtxt.Text
+ "')", con);
    cmd.ExecuteNonQuery();
    MessageBox.Show("successful");
    con.Close();
}
```

Ez a metódus akkor fut le, amikor a felhasználó rákattint az "Add" (Hozzáadás) gombra. Létrehoz egy kapcsolatot a MySQL adatbázissal (MySqlConnection), majd végrehajt egy INSERT SQL lekérdezést, hogy új rekordot szúrjon be a project táblába. A szövegdobozok értékeit (Nametxt.Text, Addtxt.Text, stb.) a felhasználó adja meg. Miután a rekordot hozzáadta, egy üzenet jelenik meg a sikeres műveletet jelzően, és az adatbázis kapcsolat lezárul.

Adatok törlése az adatbázisból (button2 Click):

```
private void button2_Click(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection("server=localhost;user
id=root;database=projectmysql");
    con.Open();
    MySqlCommand cmd = new MySqlCommand("DELETE FROM `project` WHERE
rollno = '" + Rolltxt.Text + "'", con);
    cmd.ExecuteNonQuery();
    con.Close();
}
```

```

        MessageBox.Show("successful");
    }

```

Ez a metódus akkor fut le, amikor a felhasználó rákattint a "Delete" (Törlés) gombra. Az INSERT helyett most egy DELETE SQL lekérdezést hajt végre, amely törli azt a rekordot, amelynek rollno (azonosító) mezője megegyezik a felhasználó által megadott értékkel (Rolltxt.Text).

Adatok frissítése az adatbázisban (button3 Click):

```

private void button3_Click(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection("server=localhost;user
id=root;database=projectmysql");
    con.Open();
    MySqlCommand cmd = new MySqlCommand("UPDATE `project` SET
`name`='" + Nametxt.Text + "',`address`='" + Addtxt.Text +
"',`phone`='" + Phonetxt.Text + "',`email`='" + emailtxt.Text + "'
WHERE rollno = '" + Rolltxt.Text + "'", con);
    cmd.ExecuteNonQuery();
    MessageBox.Show("successfully updated");
    con.Close();
}

```

Ez a metódus akkor fut le, amikor a felhasználó rákattint az "Update" (Frissítés) gombra. Itt egy UPDATE SQL lekérdezést hajtunk végre, amely frissíti a project táblában az adott rekordot a megadott rollno érték alapján, és a felhasználó által megadott új adatokat (Nametxt.Text, Addtxt.Text, stb.) menti el.

Űrlap törlése (button4 Click):

```

private void button4_Click(object sender, EventArgs e)
{
    Addtxt.Clear();
    emailtxt.Clear();
    Phonetxt.Clear();
    Rolltxt.Clear();
    Nametxt.Clear();
}

```

Ez a metódus akkor fut le, amikor a felhasználó rákattint a "Clear" (Törlés) gombra. Az összes szövegdobozt (pl. Nametxt, Addtxt, stb.) törli, hogy a felhasználó új adatokat adhasson meg.

Adatok megjelenítése a DataGridView-ben (gridview):

```

void gridview()
{
    MySqlConnection con = new MySqlConnection("server=localhost;user
id=root;database=projectmysql");
    con.Open();
    MySqlDataAdapter sda = new MySqlDataAdapter("select * from
project", con);
    DataTable dt = new DataTable();
    sda.Fill(dt);
    dataGridView1.Rows.Clear();
    foreach (DataRow item in dt.Rows)
    {
        int n = dataGridView1.Rows.Add();
        dataGridView1.Rows[n].Cells[0].Value =
item["rollno"].ToString();
        dataGridView1.Rows[n].Cells[1].Value =
item["name"].ToString();
    }
}

```

```

        dataGridView1.Rows[n].Cells[2].Value =
item["address"].ToString();
        dataGridView1.Rows[n].Cells[3].Value =
item["phone"].ToString();
        dataGridView1.Rows[n].Cells[4].Value =
item["email"].ToString();
    }
}

```

Ez a metódus az adatokat tölti be az adatbázisból és jeleníti meg őket a dataGridView1 vezérlőben. A MySqlDataAdapter segítségével lekérdezi az összes adatot a project táblából, majd egy foreach ciklus segítségével feltölti a DataGridView sorait.

Kattintás a DataGridView soraira (dataGridView1 MouseDoubleClick):

```

private void dataGridView1_MouseDoubleClick(object sender,
MouseEventArgs e)
{
    int n = dataGridView1.SelectedRows[0].Index;
    Rolltxt.Text = dataGridView1.Rows[n].Cells[0].Value.ToString();
    Nametxt.Text = dataGridView1.Rows[n].Cells[1].Value.ToString();
    Addtxt.Text = dataGridView1.Rows[n].Cells[2].Value.ToString();
    Phonetxt.Text = dataGridView1.Rows[n].Cells[3].Value.ToString();
    emailtxt.Text = dataGridView1.Rows[n].Cells[4].Value.ToString();
}

```

Ez a metódus akkor fut le, amikor a felhasználó duplán kattint egy sorra a dataGridView1 vezérlőben. A kattintott sor értékeit (pl. rollno, name, address, stb.) átviszi a megfelelő szövegdobozokba, hogy a felhasználó módosíthassa őket.

Önreflexió

A projekt során elsajátítottam, hogyan lehet Windows Forms alkalmazással adatokat kezelni MySQL adatbázisban. Megtanultam a hozzáadás, törlés, frissítés és megjelenítés lépéseit, valamint fejlődött a programozási készségem.