

Dunaújvárosi Egyetem Bánki Donát Technikum

Projekt Feladat Dokumentáció

Projekt tervezője: Nagy Gergő

Projekt címe: Adatbázis-kezelő

Osztály: 13.C

Bevezetés

A projekt célja egy adatbázis-alapú nyilvántartó rendszer létrehozása, amely lehetővé teszi a felhasználói adatok tárolását, módosítását és törlését. A programot C# nyelven készítettük, és MySQL adatbázist használunk az adatok kezelésére.

Felhasznált programok

MySQL – Az adatok tárolására és kezelésére.

phpMyAdmin – Az adatbázis vizuális kezeléséhez.

A program működése: A program lehetőséget biztosít az adatok felvitelére, szerkesztésére és törlésére. Az adatok a MySQL adatbázisban kerülnek tárolásra, amelyet a XAMPP Control Panel-el a phpMyAdmin felületen keresztül a megfelelő table-ben meg lehet tekinteni.

Az "Insert" gomb megnyomásával a beírt adatok bekerülnek az adatbázisba.

A "Delete" gomb megnyomásával a kijelölt rekord törlésre kerül.

Az "Update" gombbal a már meglévő rekordok szerkeszthetők.

A "New" gombbal a felhasználó kitörölheti a mezőket és új adatokat vihet be.

A program sikeres végrehajtása esetén egy felugró ablak jelenik meg a képernyőn, amely visszajelzést ad a felhasználónak arról, hogy a művelet sikeresen befejeződött. Ez segít megerősíteni, hogy az adatok megfelelően lettek mentve, módosítva vagy törölve az adatbázisból.

Ablak

Roll no: 1

Name: Gergő

Address: Dunaújváros május 1 u.

Phone: 303758332

Email: nagygergo@gmail.com

Rollno	Name	Address	Phone	Email

Insert Delete Update New

És megjelenik a táblázatba beírt adat:

Ablak

Roll no: 1

Name: Gergő

Address: Dunaújváros május 1 u.

Phone: 303758332

Email: nagygergo@gmail.com

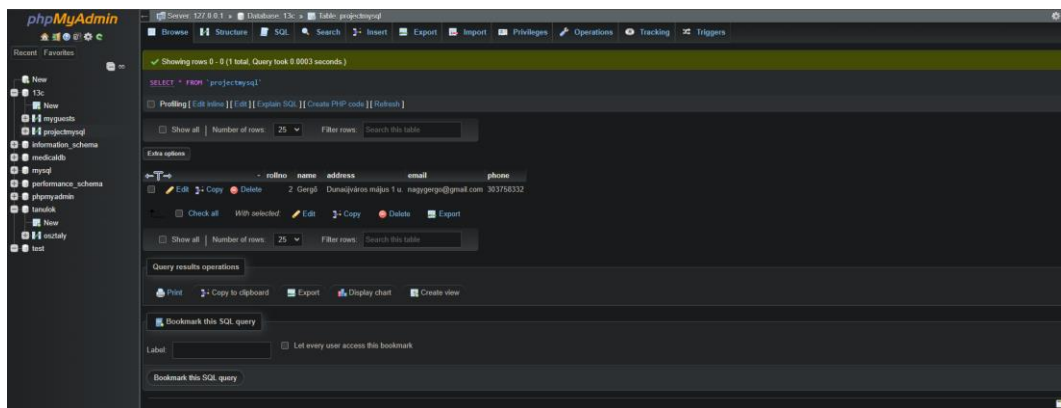
Rollno	Name	Address	Phone	Email
2	Gergő	Dunaújváros m...	303758332	nagygergo@g...

Insert Delete Update New

Adatbázis felépítése

A MySQL adatbázis egy "projectmysql" nevű táblát tartalmaz, amely az alábbi oszlopokat foglalja magába:

- **rollno** (elsődleges kulcs, egyedi azonosító)
- **name** (felhasználó neve)
- **address** (felhasználó címe)
- **phone** (telefonszám)
- **email** (e-mail cím)



A program működését igazolják a mellékelt képek, amelyeken látható a felhasználói felület, az adatok felvitele, valamint az adatbázisba történő sikeres mentés.

A projekt összegzése

A fejlesztés sikeresen elérte a kitűzött célokat, egy könnyen kezelhető és hatékony adatbázis-kezelő alkalmazást hozva létre. A rendszer lehetőséget biztosít a felhasználói adatok egyszerű tárolására, módosítására és törlésére.

A jövőbeni fejlesztések között szerepelhet egy keresési funkció beépítése az adatok gyorsabb elérése érdekében, valamint jogosultságkezelési rendszer kialakítása, amely növelné a biztonságot és a hozzáférések szabályozását.

```
{  
    public Form1()  
}
```

```

{
    InitializeComponent();
    gridview();
}

```

```

private void insert_Click(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection("server=localhost;user
id=root;database=13c");
    con.Open();
    MySqlCommand cmd = new MySqlCommand("INSERT INTO projectmysql
(`name`, `address`, `phone`, `email`) VALUES ('" + txtname.Text + "','" + txtadd.Text + "','"
+ txtphone.Text + "','" + txtemail.Text + "')", con);
    cmd.ExecuteNonQuery();
    MessageBox.Show("successful");
    con.Close();
    gridview();
}

```

```

private void delete_Click(object sender, EventArgs e)
{
    MySqlConnection con = new MySqlConnection("server=localhost;user
id=root;database=13c");
    con.Open();
    MySqlCommand cmd = new MySqlCommand("DELETE FROM `projectmysql`
WHERE rollno ='" + txtroll.Text + "'", con);
    cmd.ExecuteNonQuery();
    con.Close();
    MessageBox.Show("successful");
}

```

```

private void update_Click(object sender, EventArgs e)
{

    MySqlConnection con = new MySqlConnection("server=localhost;user
id=root;database=13c");
    con.Open();
    MySqlCommand cmd = new MySqlCommand("UPDATE `projectmysql` SET
`name`='" + txtname.Text + "', `address`='" + txtadd.Text + "', `phone`='" + txtphone.Text +
"', `email`='" + txtemail.Text + "' WHERE rollno = '" + txtroll.Text + "'", con);
    cmd.ExecuteNonQuery();
    MessageBox.Show("successfully updated");
    con.Close();
}

```

```

private void neww_Click(object sender, EventArgs e)
{
    txtadd.Clear();
    txtemail.Clear();
    txtphone.Clear();
    txtroll.Clear();
    txtname.Clear();
}

```

```

void gridview()
{
    MySqlConnection con = new MySqlConnection("server=localhost;user
id=root;database=13c");
    con.Open();
    MySqlDataAdapter sda = new MySqlDataAdapter("select * from projectmysql",
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();

}
}

private void dataGridView1_MouseDoubleClick(object sender, MouseEventArgs e)
{
    int n = dataGridView1.SelectedRows[0].Index;
    txtroll.Text = dataGridView1.Rows[n].Cells[0].Value.ToString();
    txtname.Text = dataGridView1.Rows[n].Cells[1].Value.ToString();
    txtadd.Text = dataGridView1.Rows[n].Cells[2].Value.ToString();
    txtphone.Text = dataGridView1.Rows[n].Cells[3].Value.ToString();
    txtemail.Text = dataGridView1.Rows[n].Cells[4].Value.ToString();
}
}
}

```

Önreflexió:

A program elkészítése nem volt különösen nehéz. Az adatok felvitele, szerkesztése és törlése egyszerűen megoldható volt a MySQL és phpMyAdmin segítségével, és a műveletek gyorsan megvalósíthatók voltak.

Forrás: saját forrás

