

به نام خدا

فایل توضیحات پروژه

نام پروژه : مدیریت مدرسه

استاد مربوطه : آقای احمدزاده

طراح : علی اصغر جوکار

آموزشکده فنی حرفه ای میناب

مقدمه پروژه :

این پروژه در کل دارای دو بخش می باشد:

1-پنل کاربری که شامل دانش آموزانی میشود که در سیستم ثبت نام کرده اند و از طریق فرم ورود به پنل شخصی خود وارد میشوند و در این پنل دانش آموزان قادر هستند که اطلاعات خود را در پنل مشاهده کنند و در صورت لزوم میتوانند از طریق دکمه **(ویرایش اطلاعات)** اقدام به ویرایش اطلاعات خود کنند همچنین در این پنل امکان **(مشاهده کارنامه)** و **(مشاهده حضور و غیاب)** نیز وجود دارد.

از دیگر امکانات این پنل **(حذف حساب کاربری)** دانش آموز به طور کامل از سیستم می باشد.

2-پنل مدیریت که توسط مدیران اداره میشود میتوانند اطلاعات تمامی دانش آموزانی که در این سیستم ثبت نام کرده اند را مشاهده کنند و در صورت لزوم از طریق دکمه (ویرایش) اقدام به ویرایش اطلاعات دانش آموزان از طریق جدول نمایند.

همچنین میتوانند از طریق دکمه (ثبت دانش آموز جدید) به طور دستی دانش آموز جدیدی را به سیستم اضافه نمایند. مدیران میتوانند از طریق دکمه (ثبت نمرات) نمرات سالانه هر دانش آموز را در سیستم ثبت و همچنین از طریق دکمه (حضور و غیاب) وضعیت حضور و غیاب هر دانش آموز را به طور هفتگی در سیستم ثبت کنند.

برای مدیران امکان (حذف) هر دانش آموز از سیستم به طور دستی نیز وجود دارد.

فرم خوش آمدید(Welcome):

این فرم در واقع در هنگام ورود کاربر به اپلیکیشن باز میشود و شامل دکمه های شروع و خروج است .

```
private void button2_Click(object sender, EventArgs e)
{
    Application.Exit();
}
```

با زدن دکمه خروج اپلیکیشن طبق کد بالا به طور کامل بسته میشود.

```
private void button1_Click(object sender, EventArgs e)
{
    Login login = new Login();
    this.Hide();
    login.ShowDialog();
}
```

در خط اول من نمونه ای از فرم لاگین را ایجاد کرده ام.
در خط دوم نوشته ام که بعد از ایجاد فرم لاگین فرم کنونی مخفی شود (Hide).

در خط سوم برنامه تا زمانی که فرم لاگین باز است متوقف میشود .

فرم ورود (login):

در این کاربر با وارد کردن نام کاربری و رمز عبور و زدن تیک چک باکس به شرطی که اطلاعات کاربر در پایگاه داده موجود باشد میتواند وارد حساب کاربری خود شود.

```
if (textpassword.Text != "" && textname.Text != "") // اگر مقادیر خالی نبود  
{  
    if (checkBox1.Checked) // چک باکس تایید شده بود  
    {
```

در ابتدا دو شرط مشخص کرده ایم که در صورتی که تکست باکس ها خالی نباشد و چک باکس فعال باشد..

```
OleDbConnection con = new OleDbConnection(); // ارتباط با پایگاه داده  
con.ConnectionString = "provider=Microsoft.ace.oledb.12.0;data source = DataBase90.accdb";  
con.Open();
```

در خط اول ما با پایگاه داده ارتباط برقرار میکنیم و در خط دوم نسخه پایگاه داده و نام پایگاه داده که در مسیر پروژه قرار دارد را به برنامه میدهیم و در نهایت ارتباط برقرار میشود.

```
OleDbCommand com = new OleDbCommand(); // ارتباط با جدول پایگاه داده  
com.CommandText = "SELECT COUNT(*) FROM [users] WHERE uname =? AND user_password=?";  
com.Parameters.AddWithValue("@uname", textname.Text);  
com.Parameters.AddWithValue("@user_password", textpassword.Text);  
com.Connection = con;
```

در ادامه ما در خط اول با جدول پایگاه داده ارتباط برقرار میکنیم و در خط دوم دستور انتخاب ستون های مشخص شده در جدول پایگاه داده را به برنامه میدیم که در اینجا ستون های نام کاربری و رمز عبور رو مشخص کرده ایم.

در ادامه مشخص شده است که هر کدام از ستون های جدول پایگاه داده ما مربوط به کدام تکست باکس میباشند.
و در نهایت ارتباط ما برقرار میشود.

```
if (count == 1) // اگر اطلاعات در پایگاه داده موجود بود
{
    School.Properties.Settings.Default.logged = true;
    School.Properties.Settings.Default.Save();
    School.Properties.Settings.Default.userlog = textname.Text;
    School.Properties.Settings.Default.Save();

    User_Profile form2 = new User_Profile();
    this.Hide();
    form2.ShowDialog();
    con.Close();
}
```

در اینجا ما شرطی قرار داده ایم که در صورتی که اطلاعات در پایگاه داده موجود باشد در هنگام ورود کاربر نام کاربر را بگیرد و ذخیره کند و مطمئن شود که کاربر به سیستم وارد شده است.

در این صورت فرم پروفایل کاربری ایجاد میشود و اتصال با پایگاه داده در خط آخر کد قطع میشود.

```
else if (textname.Text == "admin" && textpassword.Text == "123456")
{
    Form4 form4 = new Form4();
    this.Hide();
    form4.ShowDialog();
}
```

در ادامه شرط اول ما شرط دیگری را قرار داده ایم که در صورتی که کاربر طبق تصویر بالا با نام کاربری (ادمین) و

پسورد **(123456)** وارد شود به پنل ادمین هدایت میشود. در فرم لاگین نیز به ای موضوع اشاره شده است .

```
else if (textpassword.Text != "" && textname.Text == "")
{
    MessageBox.Show("لطفا نام کاربری خود را وارد کنید");
}
else if (textpassword.Text == "" && textname.Text != "")
{
    MessageBox.Show("لطفا رمزعبور خود را وارد نمایید");
}
else
{
    MessageBox.Show("لطفا اطلاعات خود را تکمیل کنید");
}
```

در ادامه ما دو شرط دیگر مشخص کرده ایم که در صورتی که هر کدام از تکست باکس ها حاکی بود در یک پیام به کاربر خطای مربوط را نمایش دهد.

```
private void button2_Click(object sender, EventArgs e)
{
    Sign_Up form3 = new Sign_Up();
    this.Hide();
    form3.ShowDialog();
}
```

با زدن دکمه ثبت نام فرم ثبت نام باز میشود.
با زدن دکمه خروج اپلیکیشن به طور کامل بسته میشود.

فرم ثبت نام (Sign up):

در این فرم کاربر با پر کردن اطلاعات در سیستم ثبت نام میکند و اطلاعاتش در پایگاه داده ذخیره میشود.

```
if (txtname.Text == "")
{
    MessageBox.Show("لطفا نام خود را وارد نمایید");
}
else if (txtusername.Text == "")
{
    MessageBox.Show("لطفا نام کاربری خود را وارد نمایید");
}
else if (txtpayee.Text == "")
{
    MessageBox.Show("لطفا پایه تحصیلی خود را وارد نمایید");
}
else if (txtnumber.Text == "")
{
    MessageBox.Show("لطفا شماره تماس خود را وارد کنید");
}
else if (txtpassword.Text == "")
{
    MessageBox.Show("لطفا رمز عبور خود را وارد کنید");
}
else if (txtrepassword.Text == "")
{
    MessageBox.Show("لطفا رمز عبور خود را تکرار کنید");
}
else if (!checkBox1.Checked)
{
    MessageBox.Show("در چک باکس تایید کنید که ربات نیستید");
}
else if (txtrepassword.Text != txtpassword.Text)
{
    MessageBox.Show("رمز های عبور باهم مطابقت ندارند");
}
```

در ابتدا ما تعدادی شرط مشخص کرده ایم که کاربر در هنگام ثبت نام تمامی مقادیر خواسته شده را پر کرده باشد و همچنین چک باکس تایید شده باشد و رمز عبور و تکرار آن با هم عبارتی برابر باشند.

در این صورت اپلیکیشن با پایگاه داده ارتباط برقرار میکند و بررسی میکند که آیا نام کاربری فردی که در حال ثبت نام است در پایگاه داده موجود است یا نه اگر موجود بود...

```
if (count == 1)
{
    con.Close();

    MessageBox.Show("نام کاربری در حال حاضر وجود دارد");
}
```

ارتباط با پایگاه داده قطع میشود و این پیغام را نمایش میدهد

```
OleDbConnection conn = new OleDbConnection();
conn.ConnectionString = "provider=Microsoft.ace.oledb.12.0;data source = DataBase90.accdb";
conn.Open();

OleDbCommand com = new OleDbCommand();
com.CommandText = "insert into [users]([name],[uname],[phone_number],[base0],[user_password]) values(?,?,?,?)";
com.Parameters.AddWithValue("@name", txtname.Text);
com.Parameters.AddWithValue("@uname", txtuname.Text);
com.Parameters.AddWithValue("@phone_number", txtnumber.Text);
com.Parameters.AddWithValue("@base0", txtpaye.Text);
com.Parameters.AddWithValue("@user_password", txtpassword.Text);
com.Connection = conn;
com.ExecuteNonQuery();
conn.Close();
MessageBox.Show("ثبت نام با موفقیت انجام شد");

Login form1 = new Login();
this.Hide();
form1.ShowDialog();
```

در غیر این صورت ارتباط مجدداً با پایگاه داده برقرار میشود و اینبار از دستور **درج** استفاده میکنیم و تمامی اطلاعاتی که در تکست بکس ها وجود دارد در جدول پایگاه داده و در ستون های مشخص شده ذخیره میشود.

در ادامه ارتباط با پایگاه داده قطع میشود و پیغام موفقیت امیز به کاربر نمایش میدهد.

پنل کاربری (User profile):

در این فرم کاربر امکان مشاهده اطلاعات خود را خواهد داشت همچنین امکان مشاهده نمرات و حضور و غیاب نیز از طریق همین فرم میسر خواهد بود کاربر میتواند در صورت لزوم حساب کاربری خود را به طور کامل حذف کند

```
OleDbConnection con = new OleDbConnection(); // ارتباط با پایگاه داده
con.ConnectionString = "Provider=Microsoft.ace.oledb.12.0;Data Source=DataBase90.accdb"; // ده
con.Open();
OleDbCommand com = new OleDbCommand(); // ارتباط با جدول پایگاه داده
com.CommandText = "SELECT * FROM [users] WHERE [uname]=?"; // دستور
com.Parameters.AddWithValue("@uname", School.Properties.Settings.Default.userlog);
com.Connection = con;
OleDbDataReader reader = com.ExecuteReader(); // خواندن اطلاعات از پایگاه داده
if (reader.Read())
```

در اینجا با پایگاه داده ارتباط برقرار میکنیم و در خط دوم نسخه و نام پایگاه داده را به برنامه میدهیم و در ادامه با جدول پایگاه داده ارتباط برقرار میکنیم و ستون نام کاربری را از جدول انتخاب میکنیم و نام کاربری کاربر را از جدول تنظیم شده میگیریم و با برقراری ارتباط ما دستور دیگری به برنامه میدهیم که اطلاعات را از جدول پایگاه داده بخواند

```
if (reader.Read()) // در صورت خواندن اطلاعات
{
    // نمایش اطلاعات در لیبل ها
    lbname.Text = reader["name"].ToString();
    lbpaye.Text = reader["base0"].ToString();
    lbphone.Text = reader["phone_number"].ToString();
    lbuname.Text = reader["uname"].ToString();
}
else // در غیر این صورت
{
    MessageBox.Show("خطای نامشخص");
}
```

در اینجا شرطی مشخص کرده ایم که در صورتی که توانست اطلاعات را از جدول پایگاه داده بخواند اطلاعات خواسته شده را به صورت رشته در لیبل ها نمایش دهد (طبق عکس بالا).

در صورت بروز خطا در خواندن اطلاعات پیغام خطای نا مشخص به کاربر نمایش داده میشود.

```
// حذف حساب کاربری
var result = MessageBox.Show("مقدار: اطلاعات قابل بازبینی نخواهند بود", "آیا مطمئن هستید؟؟", MessageBoxButtons.YesNo);
if (result == DialogResult.Yes)
{
    OleDbConnection con = new OleDbConnection(); // ارتباط با پایگاه داده
    con.ConnectionString = "Provider=Microsoft.ace.oledb.12.0;Data Source=DataBase90.accdb";
    con.Open();
    OleDbCommand com = new OleDbCommand(); // ارتباط با جدول پایگاه داده
    com.CommandText = "delete from [users] where [uname]=?";
    string username = School.Properties.Settings.Default.userlog.ToString();
    com.Parameters.AddWithValue("@uname", username);
    com.Connection = con;
    int count = (int)com.ExecuteNonQuery();
}
```

برای حذف حساب کاربری با کلیک بر روی دکمه مربوط به آن در ابتدا پیغامی به کاربر نمایش میدهد که آیا از انجام این کار مطمئن است؟ در صورتی که کاربر بله را انتخاب کند مجددا برنامه با ارتباط با پایگاه داده و جدول آن دستور حذف را برای کاربر مشخص شده در پایگاه داده انجام میدهد.

```
if (count == 1)
{
    con.Close();

    MessageBox.Show("حساب کاربری با موفقیت از سیستم حذف شد");
    School.Properties.Settings.Default.userlog = null;
    School.Properties.Settings.Default.Save();
    Login form1 = new Login();
    this.Hide();
    form1.Show();
}
else
{
    con.Close();

    MessageBox.Show("در حذف حساب کاربری خطایی وجود دارد");
}
```

در صورتی که ارتباط موفق بشد پیغام موفقیت آمیز به کاربر نمایش میدهد و کاربر به فرم لاگین هدایت میشود و فرم کنونی برای همیشه مخفی میشود .

در صورتی که ارتباط موفق نباشد پیغام خطا به کاربر نمایش میدهد.

ویرایش (edit):

با زدن دکمه ویرایش به فرم ویرایش هدایت میشوید و امکان ویرایش تمامی اطلاعات به غیر از نام و نام خانوادگی وجود دارد

```
string username = School.Properties.Settings.Default.userlog;
OleDbConnection con = new OleDbConnection(); // ارتباط با پایگاه
con.ConnectionString = "Provider=Microsoft.ace.oledb.12.0;Data Source=DataBase90.accdb";
con.Open();
OleDbCommand com = new OleDbCommand(); // ارتباط با جدول
com.CommandText = "Update [users] set [uname]=?,[base0]=?, [phone_number]=?, [user_password]=? WHERE [uname]=? "; // بروزرسانی
com.Connection = con;
// اجرا در ستون های انتخابی جدول پایگاه داده
com.Parameters.AddWithValue("@uname", textBox1.Text);
com.Parameters.AddWithValue("@base0", textBox4.Text);
com.Parameters.AddWithValue("@phone_number", textBox2.Text);
com.Parameters.AddWithValue("@user_password", textBox3.Text);
com.Parameters.AddWithValue("@uname", username);
int raw = (int)com.ExecuteNonQuery();
con.Close();
```

پس از ارتباط با پایگاه داده و جدول از دستور آپدیت (بروزرسانی) برای عملیات در جدول استفاده میکنیم که پس از آن اطلاعاتی که کاربر در هر کدام از تکست باکس های موجود وارد کرده است در جدول بروزرسانی خواهند شد.

(کد های این بخش با بخش های قبلی شباهت دارد)

مشاهده نمرات (Scores):

در این فرم کاربر می تواند نمرت سالانه خود را مشاهده کند
نمرات توسط مدیر در سیستم ثبت میشوند.

```
OleDbConnection con = new OleDbConnection();  
con.ConnectionString = "Provider=Microsoft.ace.oledb.12.0;Data Source=DataBase90.accdb";  
con.Open();  
OleDbCommand com = new OleDbCommand();  
com.CommandText = "SELECT * FROM [users] WHERE [uname]=?";  
com.Parameters.AddWithValue("@uname", School.Properties.Settings.Default.userlog);  
com.Connection = con;  
OleDbDataReader reader = com.ExecuteReader();
```

در این فرم ما در ابتدا با پایگاه داده ارتباط برقرار میکنیم و
نسخه پایگاه داده و نام پایگاه داده را به برنامه میدهیم
سپس با جدول پایگاه داده ارتباط برقرار میکنیم و کاربر
مورد نظر را از طریق نام کاربری آن که در جدول
(کاربران) وجود دارد انتخاب میکنیم
سپس دستور خواندن اطلاعات آن کاربر را به برنامه میدهیم

```
if (reader.Read())  
{  
    lbmath.Text = reader["math"].ToString();  
    lbscience.Text = reader["science"].ToString();  
    lbpersian.Text = reader["persian"].ToString();  
    lbritual.Text = reader["ritual"].ToString();  
    lbexercise.Text = reader["exercise"].ToString();  
    lbgeography.Text = reader["geography"].ToString();  
}  
else  
{  
    MessageBox.Show("خطای نامشخص");  
}
```

در صورتی که اطلاعات کاربر از پایگاه داده خوانده شد
(شرط).

ما می‌خواهیم که اطلاعات انتخابی را در لیبل های مشخص
شده نمایش دهد و کاربر میتواند در آن فرم هر کدام از
نمرات خود را در جلوی درس مربوط مشاهده کند.

در صورتی که خواندن اطلاعات از پایگاه داده موفقیت آمیز
نباشد پیغام خطا به کاربر نمایش داده میشود(در غیر این
صورت).

وضعیت حضور و غیاب (Attendance status):

در این فرم کاربران میتوانند وضعیت حضور و غیاب خود
را در هفته جاری مشاهده کنند .

حضور و غیاب هر هفته توسط مدیر در پنل مدیریت ثبت
میشود.

کد های این فرم کاملاً مشابه کد های فرم مشاهده نمرات
(فرم قبلی) میباشد.

پنل مدیریت (Admin panel):

خب این پنل که به طور کامل برای مدیران شخصی سازی شده است دارای آپشن هایی است که در ادامه به آنها میردازیم...

در این فرم مدیران میتوانند اطلاعات دانش آموزان را در جدول گرید ویو مشاهده کنند این اطلاعات به طور مستقیم از پایگاه داده ما نمایش داده میشوند و اطلاعاتی مانند نام کاربری رمز عبور شماره تماس و غیره را در این صفحه مدیران قادر خواهند بود مشاهده کنند.

```
private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    string str = dataGridView1.Rows[e.RowIndex].Cells["id"].Value.ToString(); // انتخاب بر اساس id
    selected_index = Convert.ToInt32(str);
}
```

کدهای بالا مربوط به جدول گرید ویو میباشد که ما این قابلیت را به جدول داده ایم که امکان انتخاب سطرهای جدول فراهم باشد.

```
private void Form4_Load(object sender, EventArgs e)
{
    dataGridView1.AutoGenerateColumns = false;
    con = new OleDbConnection("provider=Microsoft.ace.oledb.12.0;data source = DataBase90.accdb");// ارتباط با پایگاه داده
    adapter = new OleDbDataAdapter("SELECT * FROM users", con);

    dt = new DataTable();
    adapter.Fill(dt); // انتقال اطلاعات پایگاه داده به جدول گرید ویو
    dataGridView1.DataSource = dt;
    // امکان حذف و اضافه در جدول
    dataGridView1.AllowUserToAddRows = true;
    dataGridView1.AllowUserToDeleteRows = true;
}
```

کدهای بالا مربوط به فرم پنل مدیریت میباشد که ما در خط اول مشخص کرده ایم ستونهای جدول گرید ویو به صورت دستی تعریف شوند.

در خط دوم با پایگاه داده ارتباط برقرار میکنیم و نام فایل و نسخه پایگاه داده را وارد میکنیم.(مسیر پایگاه داده)

در خط سوم تمامی داده های جدول انتخاب و خوانده میشود.

در خط بعدی یک شی با نام مشخص شده برای ذخیره داده های خوانده شده از پایگاه داده میسازیم.

سپس داده هارا از پایگاه داده به جدول منتقل میکنیم.

سپس اطلاعات پایگاه داده در جدول گرید ویو نمایش داده میشود.

دو خط آخر به این معنا است که به کاربران اجازه میدهد ردیف های جدید به جدول اضافه کنند و همچنین ردیف های موجود در جدول را حذف کنند.

ثبت دانش آموز جدید(Add user):

دکمه ای تحت عنوان ثبت دانش آموز جدید در فرم پنل مدیریت وجود دارد که با کلیک کردن روی آن به فرم ثبت دانش آموز هدایت میشوید و مدیران میتوانند با پر کردن مقادیری مانند نام و نام کاربری . پایه تحصیلی . شماره تماس و رمز عبور به صورت دستی و بدون نیاز به ثبت نام کاربر خاصی دانش آموز جدیدی را به سیستم اضافه کنند.

این دکمه برای این افزوده شده است که مدیران بتوانند دانش آموزانی که به دلایلی قادر به ثبت نام نیستند را خودشان به سیستم اضافه کنند.

با زدن دکمه بازگشت کاربر به فرم پنل مدیریت باز میگردد.

کد های این فرم مانند فرم ثبت نام میباشند که در قسمت بالاتر توضیح داده شد.

دکمه ویرایش (Edit):

این دکمه یک دکمه برای انجام یک عملیات به کار رفته و با کلیک کردن بر روی آن طبیعتاً به فرمی هدایت نمیشوید.

شما با تغییر در هر یک از سلول های جدول گرید ویو برای هر کاربر (مثلاً میتوانید نام کاربری یک دانش آموز را در جدول ویرایش کنید برای ویرایش سلول مورد نظر فقط کافیت روی آن دابل کلیک کنید تا انتخاب شود سپس اقدام به ویرایش کنید) سپس با کلیک کردن بر روی دکمه ویرایش اطلاعات علاوه بر جدول گرید ویو در پایگاه داده ما نیز ویرایش میشوند.

```
private void button2_Click(object sender, EventArgs e)
{
    if (dataGridView1 != null) // اگر ویرایشی رخ داده باشد
    {
        OleDbCommandBuilder commandBuilder = new OleDbCommandBuilder(adapter);
        adapter.Update(dt); // ویرایش
        MessageBox.Show("گزینه انتخابی با موفقیت ویرایش شد");
    }
    else
    {
        MessageBox.Show("هیچ تغییری در جدول رخ نداده");
    }
}
```

در ابتدا شرطی قرار داده ایم که بررسی میکند آیا گرید ویو مقدار دهی شده است یا نه.

منظور این است که داده ها در جدول وجود دارد یا نه...

سپس به طور خودکار دستورات را بر اساس تغییرات ایجاد شده در گرید ویو تولید میکند و در پایگاه داده ذخیره میکند .
و تغییرات ایجاد شده را به منبع اصلی داده ها ارسال میکند.
اگر بروزرسانی انجام شد پیغام موفقیت آمیز به کاربر نمایش میدهد در غیر این صورت پیغام خطا نمایش میدهد.

دکمه حذف دانش آموز (Delete):

با انتخاب یک سطر از جدول گرید ویو و کلیک بر روی این دکمه دانش آموز انتخاب شده از پایگاه داده به طور کامل حذف میشود .

```
if (dataGridView1.SelectedRows.Count > 0) // اگر سطر انتخاب شده باشد
{
    OleDbConnection con = new OleDbConnection(); // ارتباط با پایگاه داده
    con.ConnectionString = "Provider=Microsoft.ace.oledb.12.0;Data Source=Database90.accdb"; // مسیر پایگاه داده
    con.Open();
    OleDbCommand com = new OleDbCommand(); // ارتباط با جدول پایگاه داده
    com.CommandText = "delete from [users] where [ID]=?"; // دستورالعمل

    com.Parameters.AddWithValue("@ID", dataGridView1.CurrentRow.Cells["ID"].Value);

    com.Connection = con;
    com.ExecuteNonQuery();
    con.Close(); // پایان
    MessageBox.Show("دانش آموز انتخابی با موفقیت حذف شد");
}
else
{
    MessageBox.Show("لطفا یک سطر را انتخاب کنید");
}
```

در ابتدا ما شرطی قرار داده ایم که بررسی میکند آیا سطری از جدول گرید ویو انتخاب شده است یا خیر

...در صورتی که سطر انتخاب شده باشد برنامه با پایگاه داده ارتباط برقرار میکند (طبق آدرس) سپس با جدول پایگاه داده ارتباط برقرار میشود.

در اینجا ما از دستور **حذف** استفاده میکنیم و از برنامه میخواهیم که از جدول (کاربران) حذف شود به شرطی که شناسه کاربر انتخاب شده و همچنین موجود باشد.

پس از پایان عملیات ارتباط با پایگاه داده را قطع میکنیم و پیغام موفقیت آمیز به کاربر نمایش میدهد.

...در غیر این صورت (سطری انتخاب نشده باشد) کاربر خطا دریافت میکند.

خروج از پنل :

```
private void button7_Click(object sender, EventArgs e)
{
    Login login = new Login();
    this.Hide();
    login.ShowDialog();
}
```

با کلیک بر روی این دکمه به فرم لاگین هدایت میشوید.

ثبت نمرات (Score registration):

در این فرم مدیران میتوانند نمرات هر درس را برای هر کدام از دانش آموزان ثبت نمایند در این فرم نمرات ثبت شده در جدول بالای فرم برای کنترل بهتر نمرات نمایش داده میشود.

مدیران با پر کردن هر تکست باکس و کلیک بر روی دکمه ثبت نمرات را در سیستم ثبت میکنند.

در این فرم امکان ویرایش سلول های جدول وجود ندارد.

```
private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    string str = dataGridView1.Rows[e.RowIndex].Cells["ID"].Value.ToString();
    selected_index = Convert.ToInt32(str);

    if (e.RowIndex >= 0 && dataGridView1.Rows[e.RowIndex].Cells.Count > 0)
    {
        try
        {
            DataGridViewRow selectedRow = dataGridView1.Rows[e.RowIndex];
            // افزودن اطلاعات به جدول گرید ویو
            txtmath.Text = selectedRow.Cells[11].Value?.ToString() ?? "";
            txtscience.Text = selectedRow.Cells[12].Value?.ToString() ?? "";
            txtpersian.Text = selectedRow.Cells[13].Value?.ToString() ?? "";
            txttritual.Text = selectedRow.Cells[14].Value?.ToString() ?? "";
            txtexercise.Text = selectedRow.Cells[15].Value?.ToString() ?? "";
            txtgeography.Text = selectedRow.Cells[16].Value?.ToString() ?? "";
        }
        catch (Exception ex)
        {
            MessageBox.Show("خطا در دریافت اطلاعات سطر");
        }
    }
}
```

کد های بالا مربوط به جدول گرید ویو در این فرم است. ابتدا مقدار سلول ستون شناسه را برای ردیف انتخاب شده گرفته و به رشته تبدیل میکند.

سپس این مقدار به یک عدد صحیح تبدیل میشود که برای شناسایی ردیف انتخاب شده استفاده میشود.

شرط موجود بررسی میکند که سطر انتخاب شده معتبر باشد و مقدار آن بیشتر از صفر باشد.

سپس ردیف انتخاب شده را در متغیر **selectedRow** ذخیره میکند.

مقدار سلول های مشخص شده با ایندکس 11 تا 16 به کنترل های متنی متصل انتساب داده میشود.

از عملگر های علامت سوال برای جلوگیری از خطا استفاده شده است.

سپس بررسی میکند که مقدار آن خالی نباشد.

اگر خطایی در این فرایند رخ دهد پیامی به کاربر نشان میدهد که به طور کلی به بروز مشکل در دریافت اشاره دارد.

```
private void button2_Click(object sender, EventArgs e)
{
    if (dataGridView1.CurrentRow != null)
    {
        try
        {
            // ترتیب نمایش اطلاعات در جدول گرید ویو
            dataGridView1.CurrentRow.Cells[11].Value = txtmath.Text;
            dataGridView1.CurrentRow.Cells[12].Value = txtscience.Text;
            dataGridView1.CurrentRow.Cells[13].Value = txtpersian.Text;
            dataGridView1.CurrentRow.Cells[14].Value = txtritual.Text;
            dataGridView1.CurrentRow.Cells[15].Value = txtexercise.Text;
            dataGridView1.CurrentRow.Cells[16].Value = txtgeography.Text;
        }
    }
}
```

برای دکمه ثبت در این فرم در ابتدا شرطی مشخص کرده ایم که در ابتدا کاربر ردیفی را از جدول گزید و یو انتخاب کرده باشد.

اگر ردیف انتخاب نشده باشد هیچ تغییری صورت نمیگیرد. مقدار سلول های ردیف انتخاب شده از گزید و یو در تکست باکس های مختلف برای نمایش تنظیم میشوند.

```
string connectionString = @"Provider = Microsoft.ACE.OLEDB.12.0; Data Source= DataBase90.accdb;";
using (OleDbConnection connection = new OleDbConnection(connectionString))
{
    string query = "UPDATE users SET [math] = @math,[science] = @science,[persian] = @persian,[ritual] = @ritual,[exercise] = @exercise ,[geography] = @geography WHERE ID = @ID";

    using (OleDbCommand com = new OleDbCommand(query, connection))
    {
        com.Parameters.AddWithValue("@math", txtmath.Text);
        com.Parameters.AddWithValue("@science", txtscience.Text);
        com.Parameters.AddWithValue("@persian", txtpersian.Text);
        com.Parameters.AddWithValue("@ritual", txtritual.Text);
        com.Parameters.AddWithValue("@exercise", txtexercise.Text);
        com.Parameters.AddWithValue("@geography", txtgeography.Text);
        com.Parameters.AddWithValue("@ID", dataGridView1.CurrentRow.Cells[0].Value);
        com.Open();
        int rowAffected = com.ExecuteNonQuery();
        if (rowAffected > 0)
        {
            MessageBox.Show("تغییرات با موفقیت ثبت شد");
        }
        else
        {
            MessageBox.Show("در ذخیره اطلاعات خطایی وجود دارد");
        }
    }
}
```

سپس با پایگاه داده ارتباط برقرار میشود. سپس مقادیر فیلد های مختلف در جدول Users بروزرسانی میشوند از @ برای مقادیر پویا استفاده شده است. سپس مقادیر پارامترهای تعریف شده در پایگاه داده با داده های فرم پر میشود.

ExecuteNonQuery برای اجرای دستور پایگاه داده استفاده میشود و تعداد رکورد های تغییر داده شده را بر میگرداند.

سپس شرطی مشخص شده است که اگر سطری ویرایش شود پیغام موفقیت نمایش داده شود در غیر این صورت به کاربر خطا دهد.

ثبت حضور و غیاب (Peresence Obsence):

در این فرم مدیران میتوانند حضور غیاب هفتگی را برای هر کدام از دانش آموزان ثبت نمایند.

مدیران با انتخاب دانش آموز از جدول و پر کردن مقادیر و سپس کلیک بر روی کمه ثبت حضور و غیاب هر دانش آموز را در سیستم ثبت میکنند.

در این فرم نمرات ثبت شده در جدول بالای فرم برای کنترل بهتر نمرات نمایش داده میشود.

کد های این فرم مشابه کد های فرم ثبت نمرات (فرم قبل) میباشد.

و این بود توضیح مختصری در رابطه با کد های پروژه و عملکرد آن.

امیدوارم که مفید بوده باشد.

توضیحات