

به نام خدا

آموزش پروژه محور

بانک اطلاعاتی در C#

استاد: حسن علیزاده

گردآورنده:

شیوا بابارشانی

دستور کار ۵ جلسه:

دانشجویان گرامی بر اساس آموزش های این جزوه نسبت به انجام پروژه خود اقدام نمایید. بخش تئوری همین جزوه بوده و بخش عملی تکمیل ۴۰ درصد از پروژه است که باید در اولین جلسه از سال ۱۳۹۸ تحویل شود.

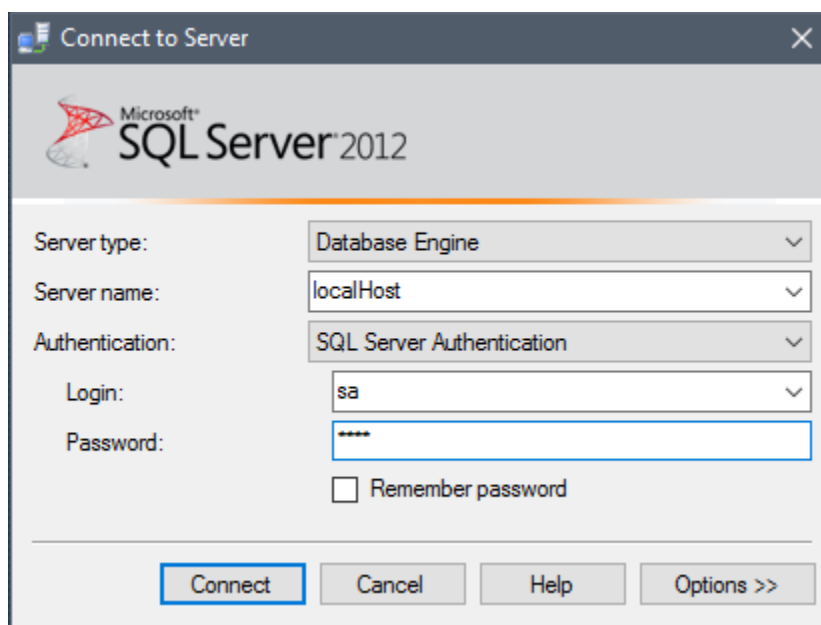
فصل اول : تعریف جداول بانک اطلاعاتی

می خواهیم نرم افزاری تحت ویندوز برای مدیریت کتاب ها، اعضا و امانات برای کتابخانه طراحی و پیاده سازی کنیم. در هر سیستم مبتنی بر بانک اطلاعات، ابتدا باید جداول لازم برای برنامه را تعریف کنیم. برای این منظور به یک سیستم مدیریت بانک اطلاعاتی نیاز داریم. سیستم های مختلفی برای این منظور وجود دارد که می توان Microsoft Access ، Microsoft SQL Server ، My SQL ، و SQL light را نام برد. ما از Microsoft SQL Server برای این پروژه استفاده می کنیم. نگاهی به سیستم های مختلفی از SQL Server وجود دارد که می توانید از هر یک از آنها استفاده کنید. ما در این پروژه از SQL Server 2012 استفاده می کنیم.

برای انجام این پروژه لازم است SQL Server بر روی سیستم شما نصب باشد. برای تهیه فایل نصبی SQL Server می توانید DVD آن را بطور جداگانه از مراکز فروش نرم افزار خریداری نمایید. یا آن را از سایتهای دانلود در اینترنت بطور رایگان دانلود نمایید. همچنین در DVD همراه این کتاب، فایل نصبی آن موجود است. در اینجا از ذکر فرآیند نصب خودداری می کنیم.

تعریف بانک اطلاعات کتابخانه

برای ایجاد بانک اطلاعات کتابخانه، نرم افزار SQL Server را اجرا کنید تا صفحه ای به شکل زیر ظاهر شود:

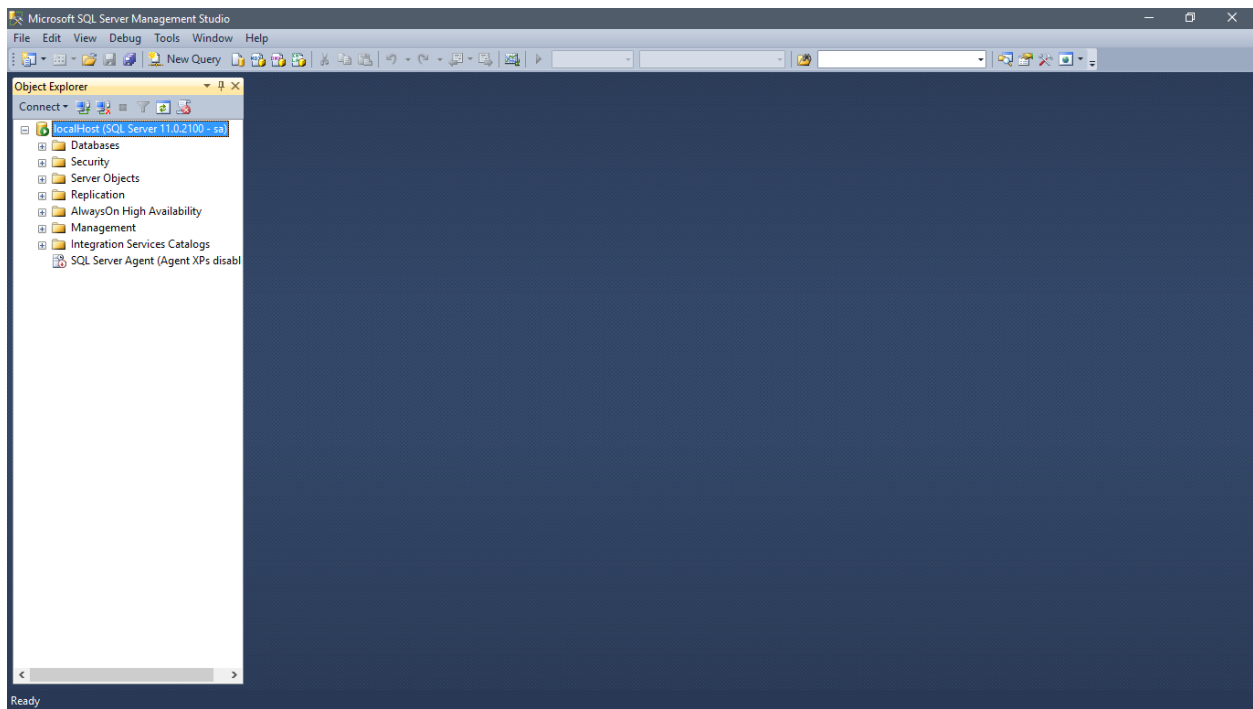


در قسمت Server name عبارت localhost و یا یک کاراکتر نقطه (.) وارد کنید که نام سرور محلی را بر روی کامپیوتر شما نشان می دهد. قسمت Authentication چگونگی احراز هویت کاربر را تعیین می کند که دو حالت دارد: Windows Authentication و SQL Server Authentication

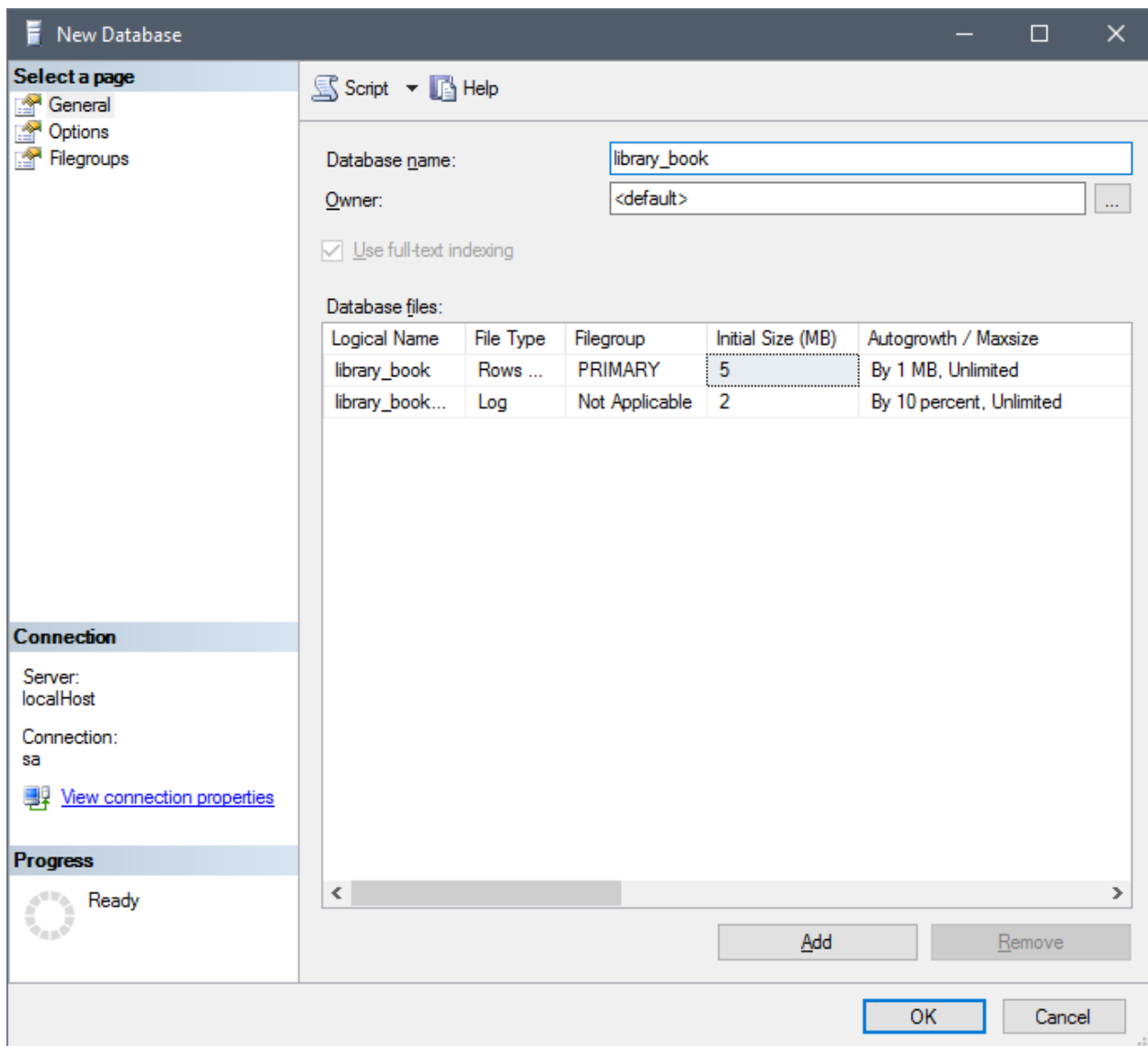
Windows Authentication: در این حالت با همان شناسه ویندوز، احراز هویت می شوید و نیازی به وارد کردن شناسه کاربری و رمز ندارید.

SQL Server Authentication: در این حالت باید شناسه و رمزی که در هنگام نصب SQL Server داده اید را وارد کنید. شناسه پیش فرض کلمه sa است. شناسه و رمز را باید به ترتیب در login و password وارد کنید.

اکنون با کلیک بر روی دکمه Connect می توانید وارد محیط SQL Server شوید. اگر همه چیز درست باشد، صفحه ای به شکل زیر برای شما باز خواهد شد:



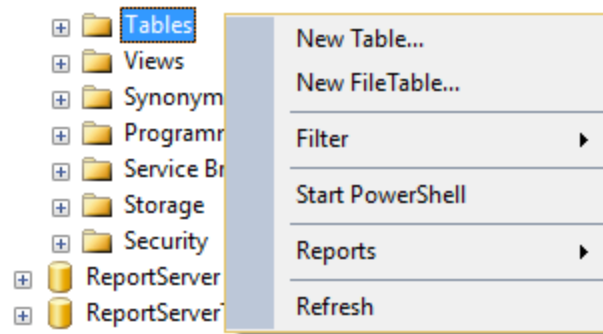
برای ایجاد بانک اطلاعات، بر روی گزینه ی Database در سمت چپ راست کلیک کرده و گزینه ی new database را انتخاب کنید تا صفحه زیر ظاهر شود:



در کادر New Database در قسمت Database name نام بانک اطلاعات خود را وارد کنید. ما نام library_book را وارد کردیم. سپس OK کنید تا بانک اطلاعات با نامی که وارد کردید، ساخته شود. اگر از پانل چپ پوشه Database را باز کنید، باید نام بانک اطلاعات ایجاد شده را ببینید.

تعریف جداول بانک اطلاعات کتابخانه

اکنون باید جداول لازم را در بانک اطلاعات کتابخانه تعریف کنیم. برای اینکار ابتدا بر روی علامت + کنار نام بانک خود کلیک کرده تا زیر مجموعه های آن باز شود سپس بر روی Table راست کلیک کرده و گزینه ی New Table را میزنیم:



برای مثال می خواهیم جدولی برای کاربران سیستم با دو ستون `password` و `username` تعریف کنیم. نام ستون و نوع آن را به ترتیب در `Column Name` و `Data Type` وارد می کنیم. چون می خواهیم این ستون ها از نوع رشته باشند، در ستون `Data Type` نوع `nvarchar(50)` را برای آنها انتخاب می کنیم. می خواهیم ستون `username` را به عنوان کلید اصلی جدول تعیین کنیم. چون نمی خواهیم شناسه کاربران تکراری باشد. برای این کار روی ستون `username` راست کلیک کرده و `Set Primary Key` را می زنیم. یک علامت کلید در کنار این ستون ظاهر می شود. این کار را از طریق نوار ابزار نیز می توانستیم انجام دهیم.

گزینه `Allow Nulls` مشخص می کند که ستون موردنظر در جدول می تواند خالی باشد یا خیر. اگر آن را تیک بزنید یعنی می تواند خالی باشد. واضح است که نمی خواهیم شناسه کاربری خالی باشد، پس تیک آن را بر می داریم.

	Column Name	Data Type	Allow Nulls
▶ 🔑	username	nvarchar(50)	<input type="checkbox"/>
	password	nvarchar(50)	<input type="checkbox"/>

اکنون جدول را ذخیره می کنیم که نام جدول از ما خواسته می شود. ما نام `tbl_user` را وارد می کنیم.

جداول کامل پروژه به شرح زیر است:

۱. جدول `tbl_user`: این جدول شامل ستون های شناسه کاربری و رمز است که کاربران برای ورود به

سیستم کتابخانه به آن نیاز دارند:

	Column Name	Data Type	Allow Nulls
▶ 🔑	username	nvarchar(50)	<input type="checkbox"/>
	password	nvarchar(50)	<input type="checkbox"/>

۲. جدول `tbl_book`: این جدول اطلاعات کتاب های موجود در کتابخانه را نگه می دارد:

	Column Name	Data Type	Allow Nulls
▶	<code>book_id</code>	int	<input type="checkbox"/>
	<code>book_name</code>	nvarchar(MAX)	<input type="checkbox"/>
	<code>book_nasher</code>	nvarchar(50)	<input type="checkbox"/>
	<code>book_writer</code>	nvarchar(50)	<input type="checkbox"/>
	<code>book_expain</code>	nvarchar(50)	<input type="checkbox"/>
	<code>tedad</code>	int	<input type="checkbox"/>
	<code>date</code>	date	<input type="checkbox"/>
	<code>id_group</code>	int	<input type="checkbox"/>

نکته: می خواهیم تمام فیلدهای `id` به طور خودکار یکی افزایش یابد. اینکار از قسمت `Column` جعبه خواص در پایین صفحه و با `yes` کردن گزینه `identity specification` انجام میگیرد.

۳. جدول `book_group`: این جدول شامل نام و شناسه گروههای موردنیاز برای دسته بندی کتاب می باشد.

	Column Name	Data Type	Allow Nulls
▶	<code>id_group</code>	int	<input type="checkbox"/>
	<code>name_group</code>	nvarchar(50)	<input type="checkbox"/>

۴. جدول `tbl_adduser`: در این جدول اطلاعات اعضایی که در کتابخانه ثبت نام میکنند را ذخیره میکنیم.

	Column Name	Data Type	Allow Nulls
▶	user_id	int	<input type="checkbox"/>
	name	nvarchar(50)	<input type="checkbox"/>
	family	nvarchar(50)	<input type="checkbox"/>
	meli	nvarchar(50)	<input type="checkbox"/>
	tell	nvarchar(50)	<input type="checkbox"/>
	jen	nvarchar(50)	<input type="checkbox"/>
	address	nvarchar(50)	<input type="checkbox"/>
	date	nvarchar(50)	<input type="checkbox"/>
			<input type="checkbox"/>


۵. جدول tbl_note: این جدول برای ذخیره ی یادداشت های کتابدار طراحی شده است.

	Column Name	Data Type	Allow Nulls
▶	id	int	<input type="checkbox"/>
	mozo	nvarchar(50)	<input type="checkbox"/>
	matan	nvarchar(MAX)	<input type="checkbox"/>

۶. جدول tbl_amanat : این جدول برای ذخیره کتاب هایی که توسط کاربر به امانت برده شده اند طراحی شده است.

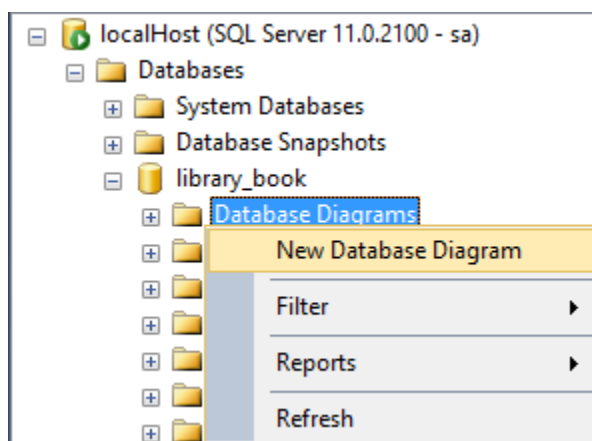
	Column Name	Data Type	Allow Nulls
▶	amanat_id	int	<input type="checkbox"/>
	user_id	int	<input type="checkbox"/>
	book_id	int	<input type="checkbox"/>
	dateamanat	date	<input type="checkbox"/>
	datereturn	date	<input type="checkbox"/>

۷. جدول tbl_giveback : در این جدول کتاب هایی که توسط کاربر به کتابخانه بازگردانده شده اند را ذخیره میکند.

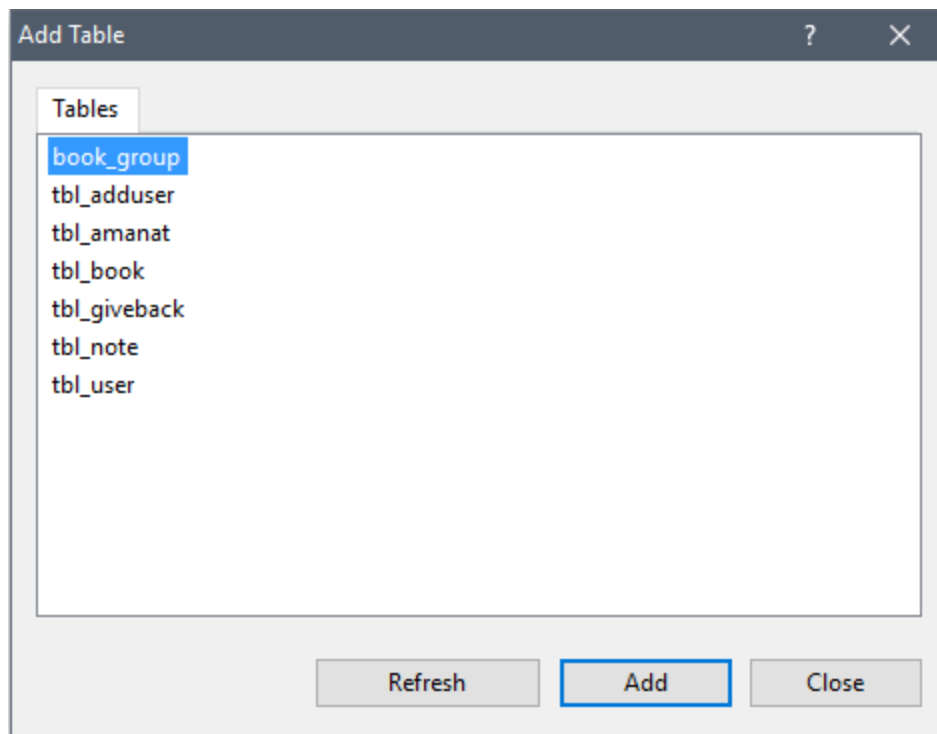
	Column Name	Data Type	Allow Nulls
	id	int	<input type="checkbox"/>
	name	nvarchar(50)	<input type="checkbox"/>
	family	nvarchar(50)	<input type="checkbox"/>
	namebook	nvarchar(50)	<input type="checkbox"/>
	m	nvarchar(50)	<input type="checkbox"/>
	modat	nvarchar(50)	<input type="checkbox"/>
	clock	nvarchar(50)	<input type="checkbox"/>
	date	nvarchar(50)	<input type="checkbox"/>

:Database Diagram

برای ارتباط جداول ما از قسمت Database Diagrams استفاده میکنیم برای اینکار مانند شکل زیر عمل میکنیم:



سپس صفحه ای مانند شکل زیر ظاهر میشود در این صفحه باید جداولی را که میخواهیم باهم ارتباط داشته باشند را انتخاب می کنیم.



پس از انتخاب صفحه ی زیر ایجاد می شود:

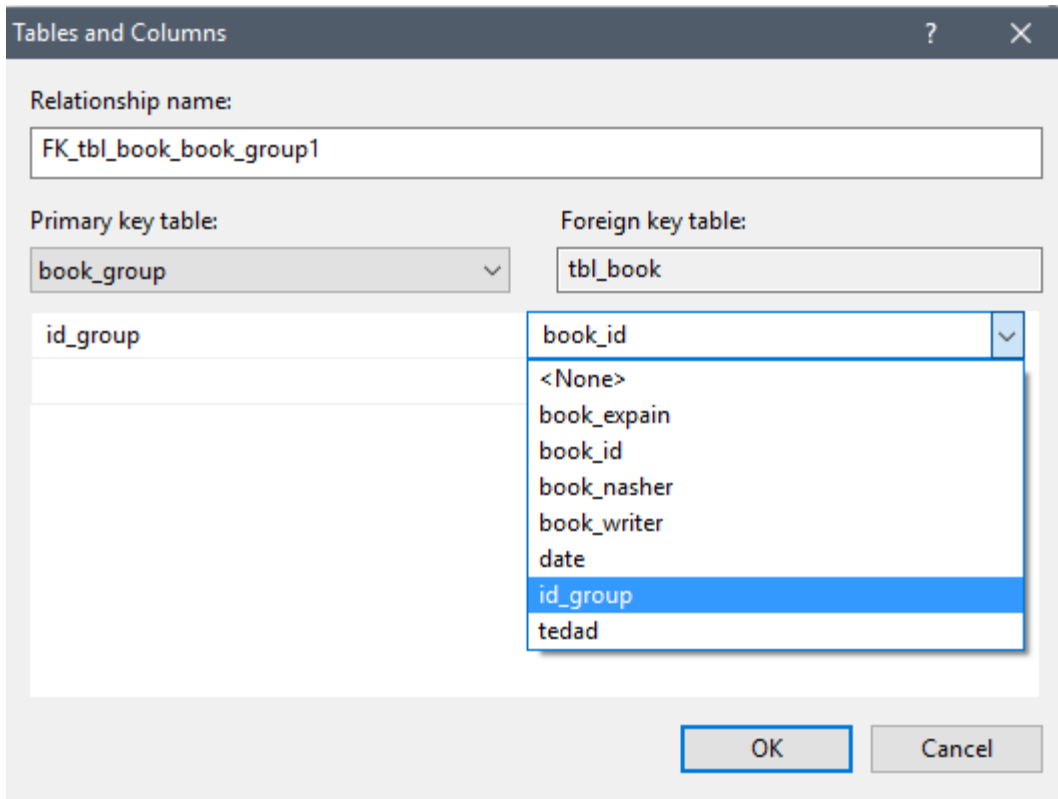
tbl_amanat *	
🔑	amanat_id
	user_id
	book_id
	dateamanat
	datereturn

book_group *	
🔑	id_group
	name_group

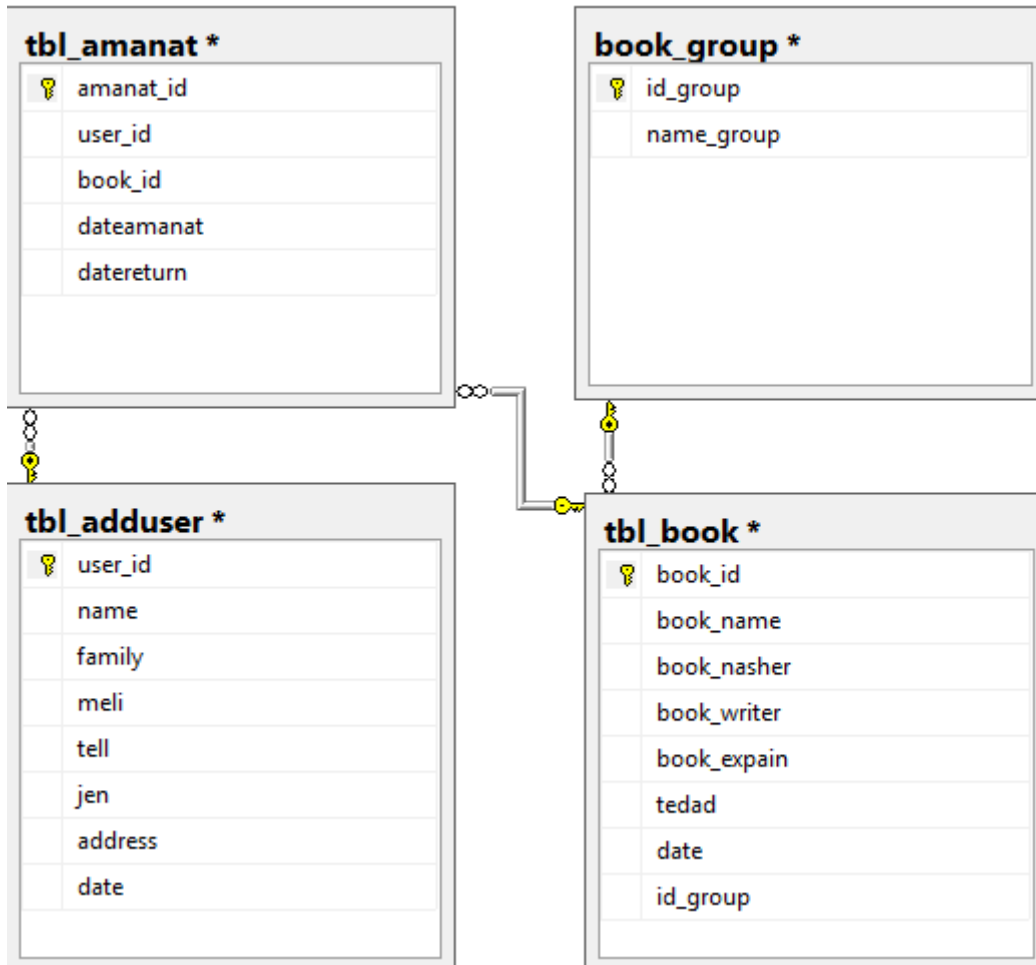
tbl_adduser *	
🔑	user_id
	name
	family
	meli
	tell
	jen
	address
	date

tbl_book *	
🔑	book_id
	book_name
	book_nasher
	book_writer
	book_expain
	tedad
	date
	id_group

برای ارتباط بین جدول tbl_group با جدول tbl_book ابتدا id_group را درگ میکنیم روی جدول Tbl_book سپس صفحه ای به شکل زیر ظاهر میشود که در اینجا باید در هر دو جدول فیلد id_group انتخاب شود.

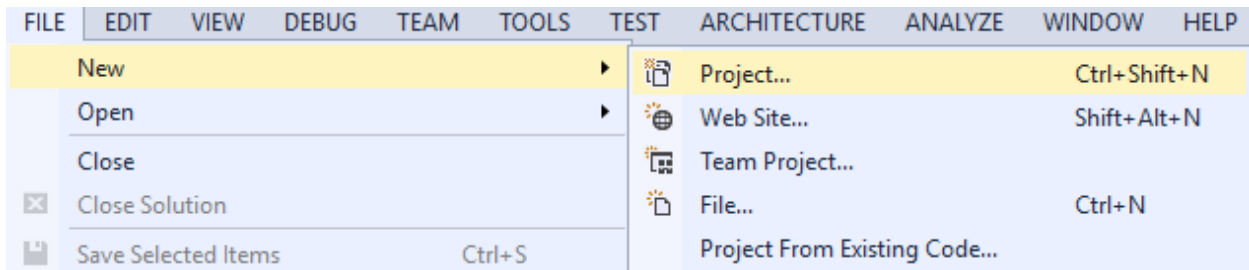


و باقی جداول را نیز مانند بالا بهم متصل میکنیم باید کلید اصلی به کلید خارجی آن در جدول دیگر ارتباط داده شود مانند شکل زیر ارتباط ها را تکمیل میکنیم.

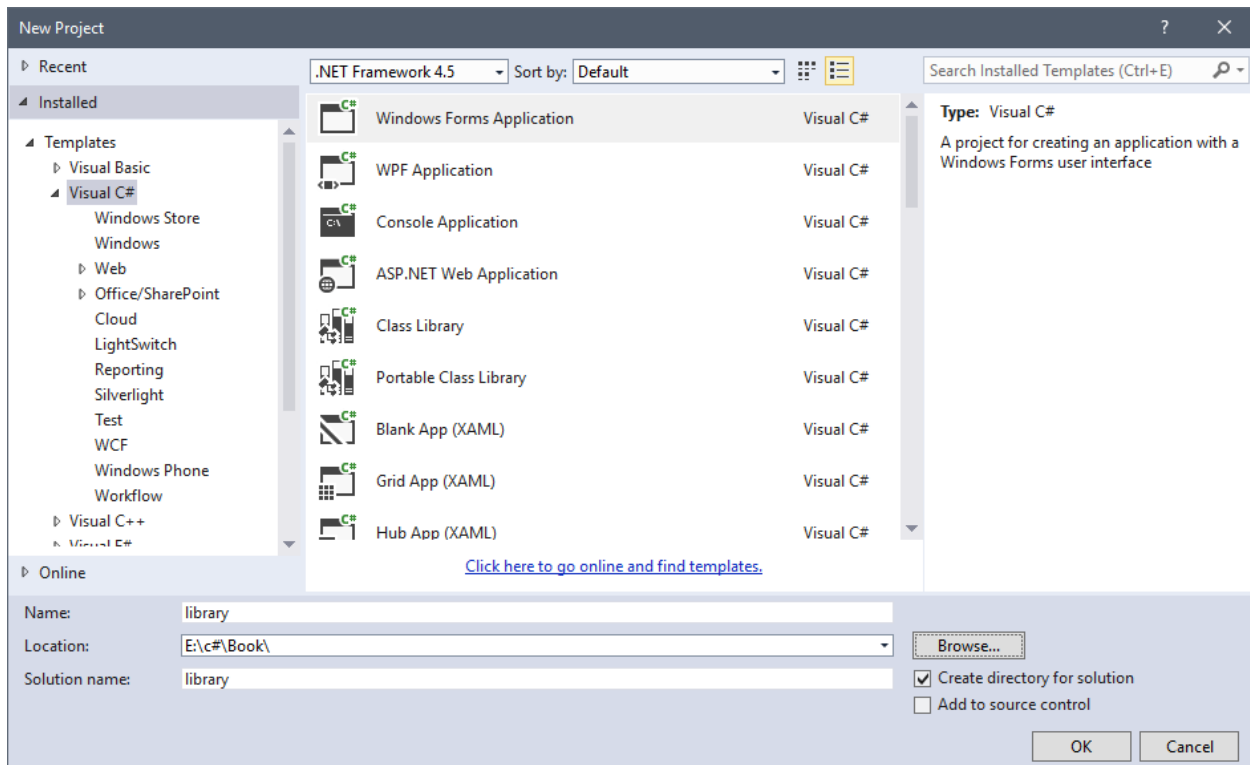


فصل دوم : ایجاد پروژه در ویژوال استودیو

ابتدا ویژوال استودیو را باز کرده و از قسمت فایل بر روی **New** و از قسمت زیر منو گزینه **New Project** را به صورت زیر انتخاب کرده



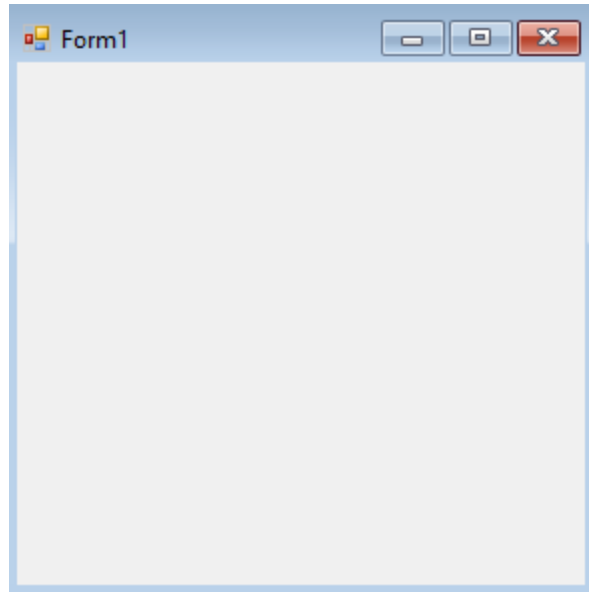
سپس صفحه **New Project** ظاهر میشود.



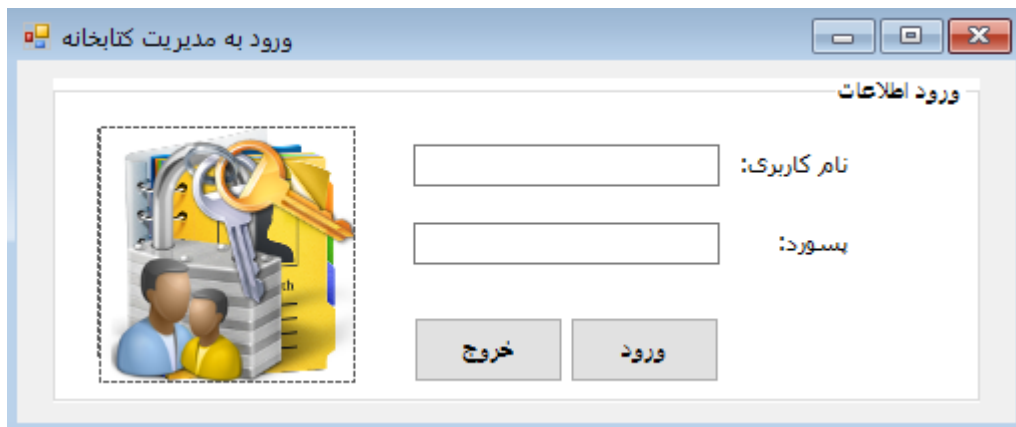
در اینجا ما از سمت راست گزینه **Visual C#** را انتخاب کرده و در سمت راست بر روی گزینه **Windows Form Application** کلیک میکنیم. در پایین صفحه در قسمت **name** نام پروژه را وارد میکنیم. ما **library** را وارد کردیم. در قسمت **Location** مسیر موردنظر خود را برای ذخیره ی پروژه انتخاب

میکنیم. در این پروژه ابتدا یک پوشه به نام book درون یک درایو دلخواه ایجاد کرده و به این پوشه آدرس می دهیم. و سپس بر روی دکمه Ok کلیک میکنیم.

با انجام عملیات بالا صفحه ای خالی به شکل زیر خواهید داشت.



با استفاده از این فرم میخواهیم فرم لاگین را به صورت زیر طراحی کنیم:



گام طراحی صفحه ی لاگین :

برای این کار ابتدا روی فرم کلیک کرده و به قسمت جعبه خواص رفته و گزینه ی Size را به مقادیر ۵۱۵، ۲۲۰ تغییر داده و در قسمت text عنوان "ورود به مدیریت کتابخانه" را تایپ میکنیم.

سپس برای زیبایی فرم به جعبه ابزار که در سمت راست صفحه قرار گرفته رفته و GroupBox را انتخاب و اندازه آن را متناسب با فرم تنظیم کرده و در جعبه خواص مقدار Text آن را "اطلاعات ورود" و Background Color آن را White قرار میدهیم.

از جعبه ابزار دو Label بر روی صفحه می کشیم و مقادیر Text آنها را به ترتیب "نام کاربری:" و "پسورد:" قرار میدهیم و مکان قرار گیری آن را همانند شکل بالا انتخاب میکنیم و همینطور از جعبه ابزار دو textbox بر روی صفحه می کشیم و مقادیر Name آنها را در جعبه خواص به ترتیب برابر با Txtuser و Ttxpassword قرار میدهیم.

برای قرار دادن آیکن از جعبه ابزار PictureBox را انتخاب میکنیم و بعد از تنظیم سایز آن وارد قسمت small panel همان ابزار میشویم. (منظور از small panel همان فلش کوچک سمت راست ابزار است) و در آنجا Choose image را انتخاب کرده و در صفحه باز شده عکس خود را انتخاب کرده و دکمه ی Import را میزنیم و بعد از قرار گیری آن در داخل Picture box برای تنظیم سایز از همان قسمت small panel از قسمت size mode گزینه ی Stretch image را انتخاب میکنیم.

برای قرار دادن دکمه های ورود و خروج نیز از جعبه ابزار button را انتخاب کرده و name آن ها را به ترتیب برابر با btnlogin و btnexit و قسمت text آن ها را "ورود" و "خروج" قرار می دهیم.

نکته: برای قرار دادن عکس داخل پروژه شما میتوانید از دو روش اقدام کنید، اول اینکه میتوانید یک پوشه درون پروژه با نام img ایجاد کرده و عکس های خود را درون آن کپی کنید و از قسمت Choose image گزینه Local resource را انتخاب کنید یا اینکه گزینه ی Project resource file را انتخاب کنید که با انتخاب این گزینه عکس موردنظر شما از درون سیستم شما انتخاب میشود.

گام کد نویسی صفحه ی لاگین:

برای اینکه بتوانیم با زدن دکمه ورود، وارد برنامه شویم، روی دکمه ی ورود دابل کلیک کرده و کدهای زیر را وارد میکنیم:

```
1. int i = 0;
2. SqlConnection con = new SqlConnection("server=localhost ;
   database = library_book ; integrated security = true");
3. SqlCommand cmd = new SqlCommand();
4. if ((Txtuser.Text == string.Empty) || (Ttxtpassword.Text ==
   string.Empty))
5. {
6.   MessageBox.Show("کنید وارد را خود عبور رمز یا کاربری نام لطفا");
7. }
8. cmd = new SqlCommand("select count(*) from tbl_user where
   username='" + Txtuser.Text + "' and password='" +
   Ttxtpassword.Text + "' ", con);
9. if (con.State == ConnectionState.Closed)
10. {
11.   con.Open();
12.   i = (int)cmd.ExecuteScalar();
13. }
14. con.Close();
15. if (i > 0)
16. {
17.   this.Hide();
18.   new frm_wait_login().Show();
19. }
20. else
21. {
22.   MessageBox.Show("است اشتباه شما پسورد یا کاربری نام");
23. }
```

توضیح قطعه کد بالا:

۱. در این برنامه ما یک متغیر برای بررسی صحت اطلاعات تعریف میکنیم.
۲. این خط برای ارتباط با بانک اطلاعاتی تعریف شده است. که ما ابتدا یک شی به نام con از کلاس SqlConnection تعریف میکنیم. مقدار localhost در پارامتر مشخص می کند که سرور sql که می خواهیم از آن استفاده کنیم در کامپیوتری قرار دارد که برنامه در آن اجرا شده است. مقدار پارامتر database نیز نام بانک اطلاعاتی است که می خواهیم به آن متصل شویم. در پایان نیز سطح دسترسی به بانک را تعیین می کنیم که این پایگاه داده برای همگان در دسترس است و یا برای فرد خاصی مورد استفاده قرار میگیرد که در این صورت مقدار user name و password را نیز تعریف می کنیم.

۳. کلاس `SqlCommand` حاوی یک دستور `sql` برای اجرا روی داده های دریافت شده از بانک اطلاعاتی است. این دستور می تواند یک دستور `select` برای انتخاب داده هایی خاص، یک دستور `insert` برای درج داده های جدید در بانک اطلاعاتی و سایر عملیات بر روی بانک اطلاعاتی باشد. در اینجا یک شی از کلاس `SqlCommand` به نام `cmd` تعریف کرده ایم تا بتوانیم از متدهای این کلاس استفاده کنیم

۴. با استفاده از این `if` ما بررسی میکنیم که آیا کاربر در `textbox` ها مقداری را وارد کرده است یا خیر.
۶. اگر مقدار آنها خالی بود با استفاده از `cd` بالا به کاربر پیغام بدهد که "نام کاربری یا پسورد خود را وارد نمایید".

۸. در اینجا ما دستوری مینویسیم که اگر مقادیری که کاربر درون `Textbox` ها وارد کرده است برابر با مقادیر فیلد های درون جدول `tbl_user` است آنها را انتخاب کن و داخل `SqlCommand` (در واقع درون شی که از نوع `SqlCommand` تعریف کرده ایم) بریزد.

۸. در این شرط ما بررسی میکنیم که آیا ارتباط ما با پایگاه داده ای که قرار است از آن استفاده بکنیم قطع است یا خیر.

۱۱. اگر ارتباط قطع بود اتصال به پایگاه داده را برقرار میکنیم.

۱۲. دستوری که در خط ۸ نوشتیم را اجرا کن و اگر مقداری که کاربر درون `Textbox` ها وارد کرده بود درون بانک اطلاعاتی موجود بود مقدار آن ستون را درون متغیر `A` بریز.

۱۴. اتصال به پایگاه داده را قطع کن.

۱۵. اگر سطر موردنظر یافت شد و کاربر رمز و نام کاربری صحیح را وارد کرده بود (در اینصورت در خط ۱۲ مقدار `A` مخالف صفر میشود) پس خط ۱۶ را اجرا کن.

۱۷. صفحه ای که اکنون در آن قرار داریم (صفحه لاگین) را ببند.

۱۸. صفحه ی بعدی را که مدنظر است را باز کن.

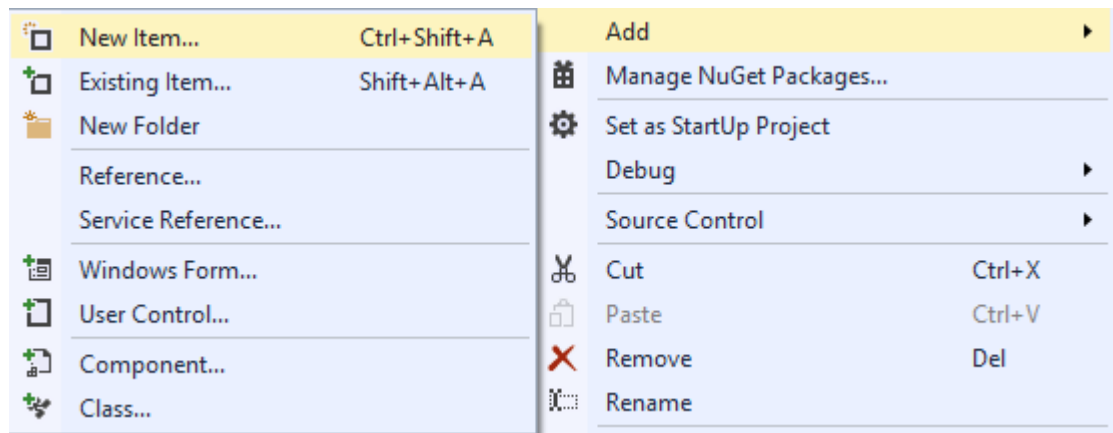
۲۱. در غیر اینصورت اگر رمز عبور و نام کاربری که کاربر وارد کرده بود اشتباه بود برو به خط ۲۲.

۲۲. به کاربر اخطار بده که "رمز عبور و یا نام کاربری شما اشتباه است".

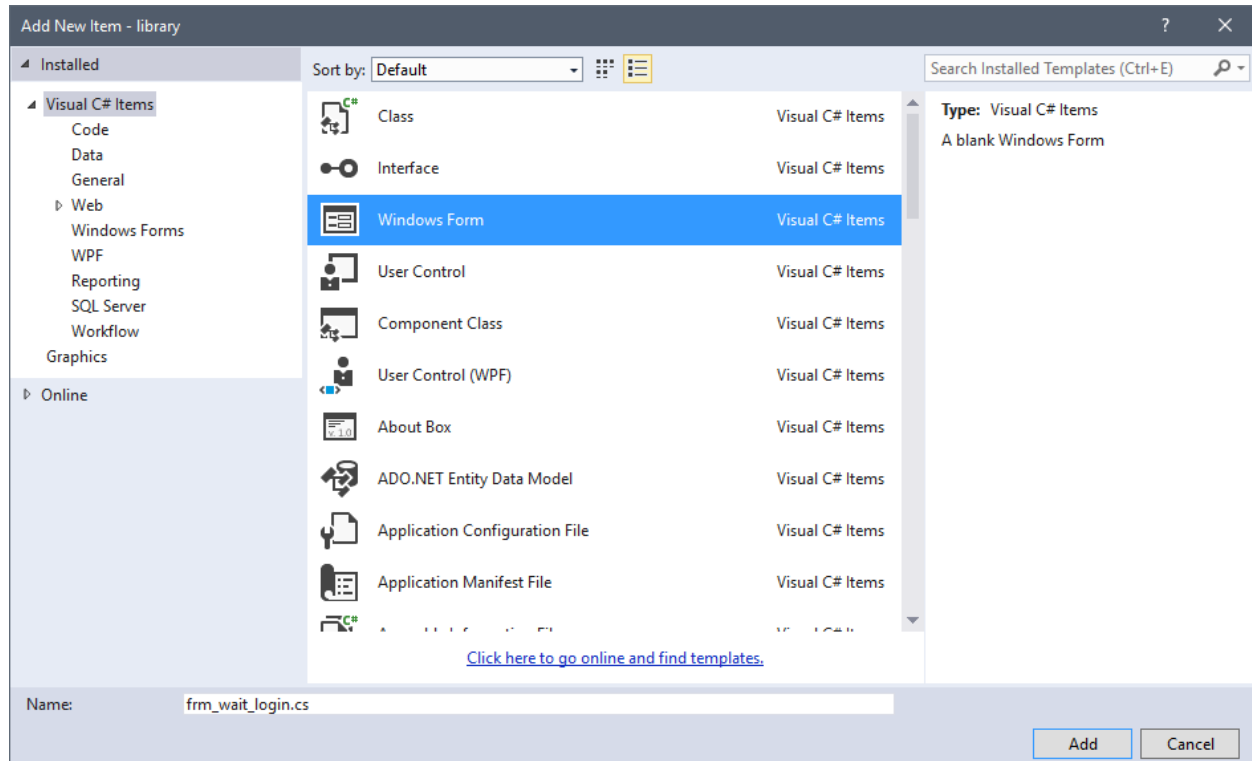
حال میخواهیم صفحه ای طراحی کنیم که با زدن دکمه ورود، صفحه ای برای انتظار جهت لود شدن صفحه ی اصلی به ما نشان دهد و پس از پر شدن جعبه پیش رونده سبز رنگ به صفحه موردنظر ما برود مانند شکل زیر:



ابتدا برای ایجاد یک پنجره جدید روی نام پروژه در سمت راست کلیک کرده و گزینه add را زده و از منوی باز شده Add New Item را همانند شکل زیر انتخاب میکنیم



پس از انتخاب این گزینه وارد صفحه ی زیر می شویم که در اینجا از سمت راست گزینه ی `visual c#` Items و از سمت چپ گزینه ی `Windows Form` را انتخاب میکنیم و سپس در قسمت `name` نام مناسبی برای فرم خود تایپ میکنیم. ما نام `frm_wait_login` را وارد کردیم. سپس بر روی گزینه ی `Add` کلیک میکنیم.



پس از ایجاد فرم تنظیمات زیر را بر روی فرم اعمال می کنیم:

`Size= 603, 477`

`StartPosition= CenterScreen`

`FormBorderStyle=None`

`ControlBox=False`

سپس ابزار `PictureBox` را روی صفحه قرار میدهیم و تصویر موردنظر خود را انتخاب کرده و سایز آن را هم با سایز فرم یکسان قرار میدهیم.

برای قرار دادن جعبه پیش رونده سبز رنگ از جعبه ابزار ابزار `ProgressBar` را انتخاب میکنیم و `name` آن را به `Proload` تغییر می دهیم. سپس از نوار ابزار یک `Timer` انتخاب کرده و از جعبه خواص مانند زیر تغییرات را بر روی آن اعمال میکنیم:

`Enabled=true`

`Interval=50`

`Modifiers=internal`

بعد از اعمال این تغییرات بروی شکل `Timer` در گوشه پایین سمت چپ پروژه دابل کلیک کرده تا وارد رویداد `Tick` شویم و در آن قطعه کد زیر را وارد میکنیم:

```
1. try
2. {
3.     Proload.Value += 1;
4.     if (Proload.Value == Proload.Maximum)
5.     {
6.         this.Hide();
7.         frm_main f = new frm_main();
8.         f.ShowDialog();
9.         timer1.Enabled = false;
10.    }
11. }
12. catch
13. {
14.    ;
15. }
```

۱: دستور `try-catch` برای اداره کردن خطاها استفاده می شود. این دستور ابتدا سعی می کند قطعه کد بخش `try` را اجرا کند، اگر موفق نشود، قطعه کدهای بخش `catch` را اجرا می کند که می تواند چاپ یک پیغام خطا باشد. ما در بخش `catch` دستوری ننوشتیم به همین خاطر فقط یک سمی کالن قرار دادیم.

۳: در این قسمت مشخص میکنیم که در چه کسری از ثانیه کادر سبز ما پر شود. اگر بجای ۱ مقدار ۱۰ را قرار دهیم متوجه خواهیم شد که سرعت پر شدن کادر خیلی سریعتر می شود.

۴ : این شرط بررسی میکند که آیا کادر پر شده یا نه. زمانی که مقدار **ProgressBar** برابر با حداکثر زمان بشود یعنی کادر کاملا پر شده است. برای تعیین مقدار حداکثر از جعبه خواص، خاصیت **Maximun** را مثلا برابر ۱۰۰ قرار می دهیم.

۵ : دستور **Hide** باعث بستن فرم فعلی می شود.

۷ : یک شی از نوع کلاس فرم ایجاد می کند.

۸ : فرم را نمایش می دهد.

۹ : **Timer** را غیر فعال می کند.

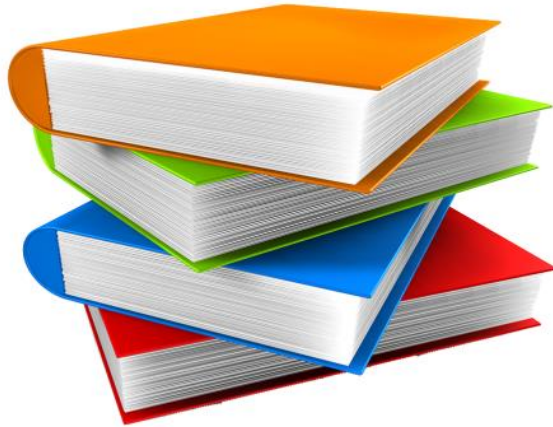
۱۲. اگر در اجرای برنامه خطایی رخ دهد، خطا را اداره می کند.

توجه: در اینجا از سه خاصیت **ProgressBar** استفاده کردیم: **Minimum, Maximum, Value**

دو خاصیت اول حداقل و حداکثر اندازه **ProgressBar** را تعیین می کنند. **Value** مقدار فعلی پیشرفت آن را نشان می دهد. دقت کنید که هر سه مشخصه از نوع صحیح هستند و **Value** حتما باید عددی بین **Minimum** و **Maximum** و یا برابر با این دو باشد.

طراحی صفحه ی اصلی:

میخواهیم صفحه ای به شکل زیر طراحی کنیم که شامل منو و زیر منو است:



پروژه مدیریت کتابخانه

سی شارپ

۱۶:۴:۵۵

۱۳۹۵/۱۱/۲۳

زیر منوی مدیریت:

مدیریت	
Ctrl+N	مدیریت کتاب
Ctrl+O	مدیریت اعضا
Ctrl+E	افزودن یادداشت

زیر منوی گزارشگیری:

گزارش گیری	
گزارش کتاب ها جدید	
گزارش امانت بردگان	

زیر منوی امکانات:

امکانات	
Ctrl+C	تغییر پسورد
Ctrl+P	پشتیبان گیری و بازگردانی

برای اینکار ابتدا طبق آموزش های بالا یک فرم جدید ایجاد کرده و سپس از جعبه ابزار **MenuStrip** را بر روی صفحه قرار داده و از جعبه خواص برای راست چین شدن `right-to-left=true` قرار میدهیم. و مقادیر را مانند شکل بالا درون کادر ها وارد میکنیم .

برای ایجاد کلید میانبر برای منوها در خاصیت **ShortcutKey** مقدار مورد نظر را انتخاب میکنیم.

برای اینکه بتوانیم با کلیک بر روی هر کدام از منوها به صفحه ی مربوطه منتقل شویم بر روی منو دابل کلیک کرده و کد رفتن به آن صفحه را می نویسی، مثلا کد رفتن به زیر منوی افزودن کتاب به این صورت است:

1. `this.Hide();`
2. `new frm_book().Show();`

۱: این صفحه جاری رو ببند.

۲: صفحه ی مورد نظر(صفحه مدیریت کتاب) را باز کن.

برای نمایش تاریخ و ساعت در این صفحه ابتدا دو **Label** روی صفحه گذاشته سپس یک **Timer** از جعبه ابزار به صفحه اضافه کرده و در جعبه خواص تنظیمات زیر را بر روی آن اعمال میکنیم:

`Enabled=true`

`Interval=100`

`Modifiers=internal`

بر روی تایمر در گوشه پایین سمت چپ دابل کلیک کرده تا وارد محیط کد نویسی شویم و کد زیر را وارد میکنیم:

1. `System.Globalization.PersianCalendar pc = new System.Globalization.PersianCalendar();`
2. `label4.Text = pc.GetYear(DateTime.Now) + "/" + pc.GetMonth(DateTime.Now) + "/" + pc.GetDayOfMonth(DateTime.Now);`
3. `label3.Text = DateTime.Now.Hour.ToString() + ":" + DateTime.Now.Minute.ToString() + ":" + DateTime.Now.Second.ToString();`

توضیح کدها:

۱: ابتدا یک شی به نام pc از نوع PersianCalender برای استفاده از تاریخ شمسی تعریف میکنم

نکته: اگر در قسمت using System.Globalization; کد using را بنویسیم دیگر لازم نیست که قبل از این کلاس کد System.Globalization را بنویسیم.

۲: سپس داخل Label مقادیر سال/ماه/روز را با فرمت و کد بالا تعریف میکنیم.

۳: در Label دوم مقدار ساعت:دقیقه:ثانیه را با فرمت و کد بالا تعریف میکنیم.

گام طراحی صفحه افزودن کتاب:

افزودن اطلاعات کتاب

کد کتاب: نام ناشر: گروه: کامپیوتر تعداد:

نام کتاب: نویسنده: توضیحات اضافه: ندارد تاریخ ثبت: ۱۳۹۵/۱۱/۲۳

افزودن ویرایش حذف صفحه اصلی

جست و جو هوشمند

کد کتاب: نام کتاب:

شناسه کتاب	نام کتاب	ناشر	نویسنده	توضیحات	گروه	تعداد	تاریخ ثبت
*							

نکته: بعد از ایجاد صفحه جدید، در قسمت بالای کد همین صفحه یعنی در قسمت using کد زیر را اضافه میکنیم:

```
using System.Data.SqlClient;
```

قطعه کد زیر برای اتصال به بانک اطلاعاتی کتابخانه است که باید از طریق تمام فرم های برنامه قابل دسترسی باشد، به همین دلیل آن را در قسمت public تعریف میکنیم:

```
SqlConnection con = new SqlConnection("server=.;database=library_book; User  
Id=sa;Password=1234");  
SqlCommand cmd = new SqlCommand();
```

متغیر `con` شیئی ای از کلاس `SqlConnection` است که توسط رشته ای به نام `ConnectionString` مقداردهی می شود. در این رشته `server` نام سرور بانک اطلاعاتی است و علامت نقطه به معنای سرور لوکال است. `Database` نام بانک اطلاعاتی است که همان `library_book` است. `User Id` و `Password` به ترتیب شناسه کاربری و رمزی که در موقع نصب `SQL Server` تعیین کردید. شناسه در حالت پیش فرض `sa` است.

توجه: اگر در هنگام باز کردن `SQL Server` از روش `Windows Authentication` استفاده کنید، نیازی به شناسه کاربری و رمز ندارید.

همچنین یک متد به نام `cleartxt` در قسمت `public` برای پاک کردن `TextBox` ها تعریف میکنیم:

```
void cleartxt()  
{  
    Txtname.Clear();  
    Txtnasher.Clear();  
    Txtwriter.Clear();  
    Txtemkanat.Text = "ندارد";  
    Txtname.Focus();  
    Txtid.Clear();  
    Txtcount.Clear();  
}
```

سپس متدی به نام `fill` برای پر کردن `DataGridView` در قسمت `public` مینویسیم:

```
1. void fill()  
2. {  
3.     SqlDataAdapter da = new SqlDataAdapter();  
4.     da.SelectCommand = new SqlCommand("select  
       book_id,book_name,book_nasher,book_writer,book_expain,tedad,date,name_group  
       from tbl_book inner join book_group on tbl_book.id_group=book_group.id_group ",  
       con);  
5.     DataTable dt = new DataTable();  
6.     da.Fill(dt);  
7.     Dvlist.DataSource = dt;  
8. }
```

توضیح متد `fill`:

۱. این متد چون مقداری باز نمیگرداند از نوع `void` تعریف میشود.

۳. یک شی از کلاس `SqlDataAdapter` با نام `da` برای اجرای دستورات در پایگاه داده تعریف میکنیم.

۴. شیء `da` دارای خاصیتی به نام `SelectCommand` است که از نوع `SqlCommand` است. همانطور که از نام آن پیداست می تواند یک دستور `Select` که به زبان `SQL` بیان می شود را به یک اتصال بانک اطلاعاتی مرتبط سازد. یعنی دو پارامتر می گیرد: یکی رشته ای است که حاوی یک دستور `Select` به زبان `SQL` است و دیگری متغیر اتصال است که قبلاً آن را با نام `con` تعریف کردیم.

۵. یک شیء به نام `dt` از کلاس `DataTable` برای ذخیره جدولی که از بانک اطلاعاتی بازیابی می شود، تعریف می کنیم.

۶. دستور `da.Fill(dt)` شیء `dt` را که یک `DataTable` است از مقادیر بازیابی شده از بانک اطلاعاتی پر می کند. `Fill` متدی از کلاس `SqlDataAdapter` است که وظیفه آن پر کردن یک `DataTable` است.

۷. اکنون که `DataTable` از داده های موردنظر پر شده است باید آنها را درون `DataGridView` نمایش دهیم. اینکار از طریق خاصیت `DataSource` از شیء `Dvlist` که یک `DataGridView` است، انجام می شود.

کد `form_load` صفحه افزودن کتاب:

```
1. fill();
2. string s = "select * from book_group";
3. SqlDataAdapter da = new SqlDataAdapter(s, con);
4. DataTable dt = new DataTable();
5. da.Fill(dt);
6. cmbgroup.DataSource = dt;
7. cmbgroup.DisplayMember = "name_group";
8. cmbgroup.ValueMember = "id_group";
```

توضیح `form_load` صفحه افزودن کتاب:

با دابل کلیک بر روی یک جای خالی از صفحه به بخش `form_load` صفحه میرویم و کد بالا را وارد میکنیم.

۱. متد `fill` را فراخوانی میکنیم تا زمانی که صفحه لود شد تمامی مقادیر موجود در `tbl_book` به کاربر نمایش داده شود.

۲. متغیر رشته ای `s` برای ذخیره یک دستور `Select` به زبان `SQL` تعریف شده است. این دستور تمام دسته بندی های کتاب را از جدول `book_group` بازیابی می کند. همانطور که به یاد دارید جدول `book_group` دسته بندی های کتاب را در خود دارد.

۳. متغیری به نام `da` از نوع `SqlDataAdapter` تعریف می کنیم و پارامترهای `s` و `con` را که قبلاً تعریف کردیم به آن می دهیم.

۴. یک `DataTable` با نام `dt` تعریف می کنیم. `DataTable` کلاسی از پیش تعریف شده است که می تواند مقادیر یک یا چند جدول را در خود نگهدارد.
۵. مقدار بدست آمده در خط ۳ را درون `DataTable` میریزیم.
۶. `cmbgroup` یک جعبه ترکیبی است که قرار است، دسته بندی های کتاب را در خود لیست کند. برای ارتباط دادن آن به جدول `dt` که یک `DataTable` از خاصیت `DataSource` آن استفاده می کنیم.
۷. اکنون باید مشخص کنیم که لیست ترکیبی باید کدام فیلد از جدول `dt` را لیست کند. این فیلد `name_group` نام دارد که قبلا در جدول `book_group` تعریف شده است.
۸. با توجه به اینکه فیلد `id_group` در همین جدول کلید است. نام آن را خاصیت `ValueMember` از لیست ترکیبی ذخیره می کنیم تا شناسه ای یکتا برای شناسایی هر عنصر از لیست داشته باشیم.

کد دکمه ی افزودن:

```

1. if (Txtname.Text != "" && Txtnasher.Text != "" && Txtwriter.Text != "" &&
   cmbgroup.Text != "" && Txtemkanat.Text != "")
2. {
3.     cmd = new SqlCommand("insert into tbl_book
   (book_name,book_nasher,book_writer,id_group,book_explain,tedad,date)values(N" +
   Txtname.Text + ",N" + Txtnasher.Text + ",N" + Txtwriter.Text + ", " +
   cmbgroup.SelectedValue + ",N" + Txtemkanat.Text + ", " + Txtcount.Text + ", " +
   Txtdate.Text + ")", con);
4.     con.Open();
5.     cmd.ExecuteNonQuery();
6.     con.Close();
7.     MessageBox.Show("گردد ثبت موفقیت با شما اطلاعات");
8.     cleartxt();
9.     fill();
10. }
11. else
12. {
13.     DialogResult msg;
14.     msg = MessageBox.Show("نمایید؟ وارد جدیدی اطلاعات میخواهید ایا نیست کامل اطلاعات",
   "Error",
   MessageBoxButtons.YesNo, MessageBoxIcon.Question,
   MessageBoxDefaultButton.Button1);
15.     if (msg == DialogResult.Yes)
16.     cleartxt();
17.     }

```

توضیح کد دکمه افزودن:

۱. در این شرط ما بررسی میکنیم که آیا کاربر **Textbox** را خالی رها کرده است یا خیر.
۳. با نوشتن این خط ما مقدار **Textbox** ها را که کاربر وارد کرده است، درون بانک اطلاعاتی ذخیره میکنیم. برای این منظور از دستور **Insert into** به زبان **SQL** استفاده می کنیم. ابتدا نام فیلدهای جدول **tbl_book** را ذکر می کنیم و در ادامه مقادیر متناظر این فیلدها را به ترتیب در قسمت **Values** می آوریم. این مقادیر همان هایی هستند که از **TextBox** ها دریافت کرده ایم. کاراکتر **N** که در ابتدای هر مقدار قرار دادیم به معنای داده یونیکد است و برای ذخیره صحیح مقادیر فارسی ضروری است. وگرنه ممکن است این مقادیر بصورت **?** نمایش داده شوند.
نکته: شکل کلی دستور **Insert into** بصورت زیر است:
Insert into tablename(filed1, filed2,...) Values (value1, value2,...)
۴. از متد **Open** برای باز کردن اتصال به بانک اطلاعات استفاده می کنیم.
۵. از متد **ExecuteNonQuery()** برای اجرای دستوری که در خط ۳ نوشتیم، استفاده میکنیم. این متد مربوط به شیء **cmd** است که از نوع کلاس **SqlCommand** است.
۶. از متد **Close()** برای بستن اتصال به بانک اطلاعات استفاده می کنیم.
۷. با استفاده از یک **MessageBox** به کاربر پیغام میدهیم که "اطلاعات با موفقیت ثبت شد."
۸. بعد از ثبت مقادیر در بانک، با استفاده از متد **cleartxt** که قبلا خودمان نوشتیم، مقادیر درون **Textbox** ها را خالی می کنیم.
۹. برای نشان دادن اطلاعات بعد از زدن دکمه ثبت متد **fill** را که قبلا نوشتیم، فراخوانی میکنیم.
۱۱. در غیر اینصورت اگر کاربر یکی از **Textbox** ها را خالی رها کرد پس کد ۱۴ را اجرا کن.
۱۳. یک متغیر از نوع **DialogResult** تعریف میکنیم.
۱۴. سپس یک اخطار به کاربر نشان میدهیم که شامل یک **title** با نام **"Error"** و دو تا دکمه **yes** و **No** و یک ایکن سوال است. این مقادیر را درون متغییری که در خط ۱۳ تعریف کردیم میریزیم.
۱۵. اگر مقداری که کاربر انتخاب کرده بود **Yes** بود.
۱۶. پس متد **cleartxt()** را فراخوانی کن.

کد دکمه حذف:

1. **string** del = Dvlist.CurrentRow.Cells[0].Value.ToString();
2. **SqlCommand** cmd = new **SqlCommand**("delete from tbl_book where book_id=@N", con);
3. cmd.Parameters.AddWithValue("@N", del);
4. con.Open();
5. cmd.ExecuteNonQuery();

6. con.Close();
7. **MessageBox.Show**("شد حذف شما موردنظر رکورد");
8. fill();

توضیح دکمه حذف:

۱. با استفاده از این کد مقدار `book_id` سطر انتخابی را درون متغیر رشته ای `del` میریزیم. `Dvlist` نام `DataGridView` و `CurrentRow` سطر فعلی است که باید حذف شود. تمام فیلدهای این سطر در آرایه ای به نام `Cells` قرار دارند که اندیس صفر، اولین فیلد از این سطر را مشخص می کند که همان `book_id` است. مقدار آن نیز با `Value` گرفته می شود. این مقدار با متد `ToString()` به رشته تبدیل می شود.

۲. متغیر `cmd` را از نوع کلاس `SqlCommand` تعریف میکنیم و یک دستور `Delete` به زبان `SQL` در آن قرار می دهیم. این دستور می گوید در جدول `tbl_book` سطر `book_id` که فیلد آن برابر پارامتر فرضی `@N` است حذف کن. این پارامتر در خط بعدی با رشته `del` که قبلا گرفتیم، مقداردهی می شود. نکته: شکل کلی دستور `Delete` در `SQL` بصورت زیر است:

شرط `Where` نام جدول `Delete From`

۳. مقدار متغیر `del` را که در خط ۱ تعریف کردیم، در پارامتر `@N` میریزیم.
۴. با متد `Open()` اتصال با بانک اطلاعاتی را باز می کنیم.
۵. با متد `ExecuteNonQuery()` دستوری که در خط ۲ نوشتیم را بر روی جدول اجرا می کنیم.
۶. با متد `Close()` اتصال با بانک اطلاعاتی را می بندیم.
۷. به کاربر پیغام می دهیم که عمل حذف با موفقیت انجام شده است.
۸. سپس متد `fill` را فراخوانی می کنیم تا `DataGridView` را مجددا نمایش دهد. این باعث می شود که سطر `حذف شده` است دیگر در `DataGridview` نمایش داده نشود.

کد دکمه ویرایش:

برای ویرایش کردن کاربر ابتدا باید بر روی سطر موردنظر دابل کلیک کند تا اطلاعات موردنظرش در هر سطر درون `Textbox`ها جای بگیرد. برای اینکار در قسمت `Design` بر روی `DataGridView` کلیک میکنیم و از جعبه خواص بر روی آیکن `event` (🖱️) کلیک میکنیم و از لیست رویدادها بر روی رویداد `RowHeaderMouseDoubleClick` دابل کلیک میکنیم تا به بخش کد این رویداد منتقل شویم. در این رویداد کدهای زیر را وارد میکنیم:

1. Txtid.Text = Dvlist.CurrentRow.Cells[0].Value.ToString();
2. Txtname.Text = Dvlist.CurrentRow.Cells[1].Value.ToString();
3. Txtnasher.Text = Dvlist.CurrentRow.Cells[2].Value.ToString();
4. Txtwriter.Text = Dvlist.CurrentRow.Cells[3].Value.ToString();
5. Txttemkanat.Text = Dvlist.CurrentRow.Cells[4].Value.ToString();
6. Txtcount.Text = Dvlist.CurrentRow.Cells[5].Value.ToString();
7. Txtdate.Text = Dvlist.CurrentRow.Cells[6].Value.ToString();
8. Txtdate.Enabled = false;
9. cmbgroup.Text = null;
10. cmbgroup.SelectedText = Dvlist.CurrentRow.Cells[7].Value.ToString();

توضیح کدهای RowHeaderMouseDoubleClick:

دستورات بالا مقادیر فیلدهای سطری که کاربر انتخاب کرده است را به ترتیب در کادرهای ویرایش متناظر بر روی فرم قرار می دهد تا کاربر بتواند آنها را ویرایش کند. **CurrentRow** همان سطر فعلی است که کاربر آن را انتخاب کرده است. آرایه **Cells** آرایه ای است که فیلدهای این سطر به ترتیب در آن ذخیره شده اند و اندیس آنها از صفر شروع می شود. **Value** مقدار درون هر عنصر از این آرایه است که توسط متد **ToString()** به رشته تبدیل می شود.

فقط توجه نمایید که در خط ۸ کادر ویرایش **Txtdate** را غیرفعال کردیم تا کاربر نتواند آن را ویرایش کند. چون تاریخی است که این رکورد در جدول ثبت شده است و نباید تغییر کند.

cmbgroup همان لیست ترکیبی است که دسته بندی های کتاب را لیست می کند و با خاصیت **SelectedText** مقداردهی می شود.

کد دکمه ویرایش:

1. **SqlCommand** cmd = new **SqlCommand**("update tbl_book set book_name=N" + Txtname.Text + ",book_nasher=N" + Txtnasher.Text + ",book_writer=N" + Txtwriter.Text + ",id_group=" + cmbgroup.SelectedValue + ",book_expain=N" + Txttemkanat.Text + ",tedad="+Txtcount.Text+" where book_id=" + Txtid.Text + """, con);
2. con.Open();
3. cmd.**ExecuteNonQuery**();
4. con.Close();
5. **MessageBox.Show**("شد ویرایش موفقیت با شما اطلاعات");
6. fill();

۱. متغیری به نام cmd از نوع کلاس SqlCommand تعریف می کنیم و یک دستور update به زبان SQL به آن می دهیم.

نکته: شکل کلی دستور update از زبان SQL به صورت زیر است:

نام جدول update

مقدار=فیلد ,مقدار=فیلد set

شرط where

کاراکترهای N در این دستور به معنای یونیکد است و برای مقادیر فارسی بکار می رود.

۲. با متد Open() اتصال با بانک اطلاعاتی را باز می کنیم.

۳. با متد ExecuteNonQuery() دستوری که در خط ۱ نوشتیم را بر روی جدول اجرا می کنیم.

۴. با متد Close() اتصال با بانک اطلاعاتی را می بندیم.

۵. به کاربر پیغام می دهیم که اطلاعات با موفقیت ویرایش شد.

۶. با فراخوانی متد fill() که قبلا نوشتیم، داده های DataGridView را بهنگام می کنیم تا نتایج تازه را نمایش دهد.

کد دکمه صفحه اصلی:

این دکمه برای بستن صفحه ویرایش و بازگشت به صفحه اصلی برنامه است.

```
this.Hide();  
new frm_main().Show();
```