

برنامه نویسی مبتنی بر وب



جناب استاد سروش

تهیه کننده: حامد گلامرچ الیاسی

Hamed_golamraj_elyasi@yahoo.com

صفحه	عنوان
	مفاهیم اولیه
۳	انواع صفحات وب
	مراحل ساخت و راه اندازی یک سایت
	ابزارهای نرم افزاری طراحی صفحات وب
	قالب استاندارد و معمول سایت
۴	سبک های بروز رسانی
	Master Pages
۶	کنترل های Validation
۷	انتقال اطلاعات بین صفحات
۹	پایگاه داده
۱۴	روش انتقال و نسخه برداری از فایل پایگاه داده SQL
۱۵	سطح دسترسی در SQL
۱۵	ارتباط با پایگاه داده از طریق Wizard
۱۹	اعمال فیلتر بر روی داده های در GridView
۲۰	WebConfig
۲۱	درج تصویر و پیوند در GridView
۲۳	AJAX
۲۶	آپلود و دانلود فایل
۲۶	XML

مفاهیم اولیه

در حالت کلی سایت از صفحه خانگی (Home Page) و صفحات فرعی (Web Pages) تشکیل شده است. پسوند های رایج صفحات وب HTML-HTML-PHP-ASP-ASPX-JSP هستند که بستگی به ابزار ساخت صفحات و سایت دارد.

- به صفحه اصلی (اول) سایت، صفحه خانگی (Home Page) گفته می شود.

هر آدرس سایت (URL) به صفحه خانگی آن سایت ختم می شود و صفحه خانگی فایلی با نام Index یا Default است و پسوند آن همانطور که در قبل گفته شد بستگی به ابزار ساخت صفحه یا سایت دارد. برای مثال صفحه خانگی سایتی که توسط ASP.NET طراحی می شود Default.aspx است. یا صفحه اول سایتی که با PHP طراحی میشود Default.php یا Index.php است.

- هر صفحه وب از دوبخش قالب و محتوا تشکیل شده است.

انواع صفحات وب

۱- صفحات ایستا (Static) : دارای محتوای ثابت.

۲- صفحات پویا (Dynamic) : دارای تعامل (تعاملی) و رویداد گرا.

مراحل ساخت و راه اندازی یک سایت

۱- طراحی وب سایت با استفاده از ابزار های مورد نظر .

۲- اجاره دامنه (مثلا tcrsadra.ir).

۳- اجاره فضای هاست برای آپلود سایت طراحی شده.

۴- معرفی به موتور های جستجو.

× ابزارهای نرم افزاری طراحی صفحات وب

وب سرور	زبان برنامه نویسی	تکنولوژی	DBMS (دیتا بیس قابل اجرا)
۱- مبتنی بر Windows Server (وب سرور IIS)	تمام زبانهای تحت پلتفرم .Net (مثل C#,VB,J#,C++)	Asp , Asp.net , Ajax	Access , SQL Server , Oracle , My SQL , DB2

• تکنولوژی مجموعه ای از Object ها است.

• سرور های مبتنی بر لینوکس محصولات مایکروسافت را پشتیبانی نمی کنند.

وظیفه وب سرور در یافت قالب و محتوا از محل های (دیتا بیس ها) مختلف و در آمیختن آنها و تبدیل آنها به کد های Html یا XML و سپس ارسال کدهای تولید شده به سمت مرورگر کاربر است.

دامنه و زیر دامنه: برای مثال yahoo.com یک دامنه است در حالی که shopping.yahoo.com یک زیر دامنه است.

تنها صفحه در Root هاست: صفحه خانگی (فایلی با نام Default یا Index) است.

به طور معمول دایرکتوری ریشه فضای هاست شامل:

Default.aspx (صفحه خانگی)

Images (پوشه حاوی عکس ها)

Pages (پوشه حاوی صفحات)

DB (پوشه حاوی دیتا بیس)

همانطور که مشاهده می شود تنها صفحه موجود در ریشه، فایل صفحه خانگی است.

قالب استاندارد و معمول سایت

Logo
Menu
Content
Copy Right

در این درس از قالب استاندارد فوق جهت طراحی سایت استفاده می شود.

سبک های بروز رسانی

Desktop : مخصوص صفحات استاتیک

Web Based : مخصوص صفحات داینامیک

هاست رایگان جهت تمرینات کلاسی : www.aspspider.net

روش های ارسال سریعتر فایل به برخی هاست ها از جمله هاست فوق:



Zip کردن سایت نهایی تولید شده، و ارسال فایل زیپ شده از طریق پنل کنترل سایت، و سپس کلیک بر روی گزینه Extract است.

تمرین: قالب یک سایت معتبر را شبیه سازی کنید و در هاست رایگان آپلود کنید.

راهنمایی:

- برای ایجاد صفحه جدید در صفحه Solution Explorer کلیک راست و سپس Add New Item\ Web Form را انتخاب کنید.
 - برای ایجاد قالب بندی کلی از جدول (Table\Insert Table) استفاده می شود.
 - برای قرار دادن لینک از کنترل Hyperlink استفاده می شود. (خاصیت NavigateUrl جهت آدرس دهی صفحه مورد نظر است)
 - برای قرار دادن تصویر از کنترل Image استفاده میشود. (خاصیت imageUrl برای آدرس دهی عکس مورد نظر است)
- سایر خصوصیات کنترل های فوق بررسی شوند.

Master Pages

مستر پیج صفحه قالب کلی سایت است، به بیان ساده تر قالب سایت در مسترپیج طراحی می شود و سایر صفحات موجود، طراحی مسترپیج را به خود می گیرند با این اوضاع در مستر پیج قالب کلی و در سایر صفحات، تنها محتوای آن صفحات قرار می گیرد. پس لازم نیست در ساخت وب سایت برای هر صفحه طراحی داشت بلکه یک قالب درون مستر پیج طراحی می شود و سایر صفحات این قالب کلی را به خود می گیرند بنابراین پس از طراحی مستر پیج تنها کفایست در هر صفحه محتوای همان صفحه را اضافه کرد و نیازی به کپی برداری ممتد قالب سایت نداریم.

روش افزودن مستر پیج:

- برای ایجاد مستر پیج در صفحه Solution Explorer کلیک راست و سپس Add New Item\ Master Page را انتخاب کنید.
- طراحی مستر پیج (مانند قرار دادن جدول، عکس منو و...) کاملاً شبیه به طراحی صفحات معمولی است.

بخش کلیدی مستر پیج

در تعریف گفته شد که صفحات، شکل مستر پیج را به خود می گیرند و تنها محتوای هر صفحه، متفاوت از مستر پیج است حال برای مشخص کردن بخش محتوا، از کنترل ContentPlaceholder استفاده می شود، کافی است کنترل ذکر شده را در بخش دلخواه مستر پیج درگ کنید.

ست کردن مستر پیج بر روی یک صفحه

بر روی صفحه مورد نظر خود کلیک کرده و در قسمت Properites خاصیت MasterPageFile را برابر با آدرس و نام مستر پیج مورد نظر خود قرار دهید.

تمرین: با استفاده از Master Page یک سایت موجود در اینترنت را شبیه سازی و آپلود کنید.

کنترل های Validation

این کنترل ها از بخش Toolbox\Valdation قابل دسترسی هستند.

-۱

CompareValidator: جهت مقایسه سازی (محتویات جعبه متن) مورد استفاده قرار می گیرد. (عموما برای چک کردن گذرواژه)

خواص مهم:

- **ControlToCompare**: نام کنترل (جعبه متن) مقایسه شونده اولیه (مقایسه شود با...)
- **ControlToValidate**: نام کنترل (جعبه متن) مقایسه شونده ثانویه.
- **ErrorMessage**: متن پیغام خطایی که در صورت یکسان نبودن محتویات جعبه متن های مشخص شده در دو خاصیت بالا می بایست نمایش داده شود.

-۲

RegularExpressionValidator: جهت اطمینان از صحیح بودن عبارات قانونمند خاص مورد استفاده قرار می گیرد (عبارات قانونمند مثل: آدرس ایمیل، شماره تلفن، آدرس سایت، کد پستی و...)

خواص مهم:

- **ControlToValidate**: نام کنترل (جعبه متن) مورد بررسی.
- **ValidationExpression**: انتخاب فرمت قابل قبول.
- **ErrorMessage**: متن پیغام خطایی که در صورت رعایت نشدن قاعده، می بایست نمایش داده شود.

-۳

RequiredFieldValidator: جهت الزامی کردن پر نمودن محتوای جعبه متن مورد استفاده قرار می گیرد.

خواص مهم:

- **ControlToValidate**: نام کنترل (جعبه متن) مورد بررسی.
- **ErrorMessage**: متن پیغام خطا.

-۴

RangeValidator: جهت محدود کردن محتوا در محدوده خاص.

خواص مهم:

- **ControlToValidate:** نام کنترل (جعبه متن) مورد بررسی.
- **Type:** نوع مورد بررسی (صحیح یا رشته ای یا....)
- **MinimumValue:** کوچکترین مقدار قابل قبول.
- **MaximumValue:** بزرگترین مقدار قابل قبول.
- **ErrorMessage:** متن پیغام خطا.

نکته: در تعیین نوع مورد بررسی و کوچکترین مقدار و بزرگترین مقدار باید دقت نمود، برای مثال در صورت انتخاب نوع صحیح، کوچکترین مقدار و بزرگترین مقدار باید حتما صحیح باشند و یا در صورت انتخاب نوع رشته ای، مقادیر بالا و پایین می بایست رشته ای باشند، همچنین مقادیر بالا حتما باید بزرگتر و یا مساوی مقادیر پایین باشند در غیر اینصورت برنامه با خطا مواجه خواهد شد.

-۵

ValidationSummary: این کنترل جهت نمایش پیام های کلیه Validation ها بصورت مرتب و یکجا است.

انتقال اطلاعات بین صفحات

به دو روش صورت پذیر است.

-۱

QueryString: دارای سرعت بالا ولی بدون هیچگونه امنیت است.

کار با روش **QueryString:**

ارسال: در صفحه ارسالی و در رویداد کلیک دکمه دلخواه به روش زیر عمل می شود:

```
Response.Redirect("&متغییر ارسالی دوم="+ داده اول + "?آدرس صفحه مقصد");
```

(مثال)

```
protected void Button1_Click(object sender, EventArgs e)
{
    Response.Redirect("~/pages/page2.aspx?id=" + TextBox1.Text + "&pass=" + TextBox2.Text);
}
```

نکته: در اعلام مسیر بجای \ از \\ استفاده میشود.

دریافت: در صفحه مقصد و در بخش Page_Load به روش زیر عمل شود:

```
متغییر دریافتی همانم با متغییر ارسالی اول = Request.Params["متغییر دریافتی همانم با متغییر ارسالی اول"];
متغییر دریافتی همانم با متغییر ارسالی دوم = Request.Params["متغییر دریافتی همانم با متغییر ارسالی دوم"];
```

توجه شود که متغییر دریافتی و متغییر ارسالی، نیازی به هیچگونه تعریف متغییر به صورت معمول ندارد!

(مثال)

```
protected void Page_Load(object sender, EventArgs e)
{
    Label1.Text = Request.Params["id"];
    Label2.Text = Request.Params["pass"];
}
```

۲-

Session: دارای امنیت بالاتری است و داده ها در هدر بسته ها قرار می گیرند.

کار با روش Session

ارسال: در صفحه ارسالی و در رویداد کلیک دکمه دلخواه به روش زیر عمل می شود:

```
Session.Add("داده اول", "متغییر ارسالی اول");
Session.Add("داده دوم", "متغییر ارسالی دوم");
Response.Redirect("آدرس صفحه مقصد");
```

(مثال)

```
protected void Button1_Click(object sender, EventArgs e)
{
    Session.Add("id", TextBox1.Text);
    Session.Add("pass", TextBox2.Text);
    Response.Redirect("~/Pages/Page2.aspx");
}
```

دریافت: در صفحه مقصد و در بخش Page_Load به روش زیر عمل شود:

```
متغییر دریافتی همانم با متغییر ارسالی اول = Session["متغییر دریافتی همانم با متغییر ارسالی اول"].ToString();
متغییر دریافتی همانم با متغییر ارسالی دوم = Session["متغییر دریافتی همانم با متغییر ارسالی دوم"].ToString();
```


(مثال)

```
protected void Page_Load(object sender, EventArgs e)
{
    Label1.Text = Session["id"].ToString();
    Label2.Text = Session["pass"].ToString();
}
```

پایگاه داده

ارتباط با پایگاه داده دارای دور روش است : Wizard و برنامه نویسی.

کار با پایگاه داده از طریق برنامه نویسی:

ایجاد ارتباط با پایگاه داده شامل چهار بخش مهم زیر است:

۱- Connection String

۲- Query String

۳- Query Execute

۴- کنترل ها

۵- هوشمندی سیستم

در ادامه این موارد شرح داده خواهند شد.

-۱

Connection String: (رشته اتصال) رشته ایست که نوع پایگاه داده، ورژن، محل فایل پایگاه داده، و امنیت، و... را مشخص می کند. رشته اتصال پایگاه داده اکسس بصورت زیر است:

"provider=microsoft.jet.oledb.4.0;data source= *آدرس فایل پایگاه داده*"

Provider در اس.کیو. ال SQLOLEDB است

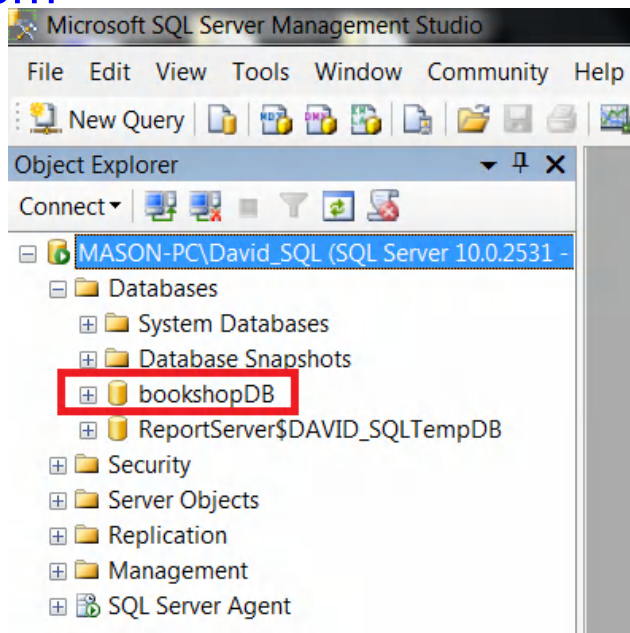
نمونه ای از رشته اتصال پایگاه داده SQL

```
string cstr = "provider=SQLOLEDB;Data Source=.;Initial Catalog=bookshopDB;Integrated Security=SSPI";
```

Data Source=. یعنی سیستم جاری (IP و Computer Name سیستم جاری)

Initial Catalog نام دیتا بیس است که با ورود به محیط اسکيوال قابل دریافت است.

برای مثال فوق:



-۲

Query String: رشته ایست که تقاضا (پرسوجو) در آن قرار می گیرد و سپس در پایگاه داده اعمال می شود.

الف) خواندن:

"Select * from *نام جدول* where *نام فیلد* = مقدار and *نام فیلد* = مقدار"

مثال)

string querystr = "select * from users where username='" + TextBox1.Text + "' and password='" + TextBox2.Text + "'";
 نکته: برای مشخص کردن رشته ای درون رشته ای دیگر از ' استفاده می شود، در صورتی که فیلد مورد نظر از نوع عددی باشد به ' نیازی نیست.

ب) درج

"insert into *نام جدول* (*اسامی فیلد ها*) Values(*مقادیر فیلدها*)"

مثال)

```
qstr="insert into users(username,password,name) values('" + TextBox1.Text + "','" + TextBox2.Text + "','" + TextBox3.Text + "')";
```

ج) حذف

"delete from *نام جدول* where *نام فیلد* = مقدار"

مثال)

```
querystr = "delete from users where username='" + TextBox1.Text + "'";
```

هشدار: در عمل حذف کردن، نام فیلد به طور معمول کلید است در غیر اینصورت تمامی فیلد هایی که برابر با مقدار داده شده هستند حذف خواهند شد برای مثال دستور زیر (با فرض غیر کلید بودن فیلد fname) تمامی افراد موجود در جدول که دارای نام Ariya هستند را حذف می کند!

```
" Delete from users where fname='ariya' ";
```

(د اصلاح

"مقدار=نام فیلد where مقدار=نام فیلد set نام جدول update"

(مثال

```
qstr = "update users set password='" + txt2 + "',name='" + TextBox3.Text + "'" + "where username='" + TextBox1.Text + "'";
```

مثال بالا مقدار متغییر txt2 ، و نام موجود در TextBox3 را برای رکوردی ست می کند که فیلد username آن معادل مقدار موجود در TextBox1 است.

هشدار: در عمل اصلاح، نام فیلد مشخص کننده مکان (فیلد پس از where)، معمولا کلید است در غیر اینصورت رکورد تمامی فیلد ها با مقدار داده شده ویرایش خواهند شد، برای مثال دستور زیر (به فرض کلید نبودن فیلد fname) فیلد سن تمامی افرادی که نامشان Mehri است را ویرایش می کند!

```
"update users set age="+ newage + " where fname='mehri'";
```

توجه: با توجه به انتقال اطلاعات در بین صفحات، عمل اصلاح نباید در صفحه درج و حذف صورت گیرد.

مثال: برنامه ای که نام کاربری و رمز عبور را به صفحه ای ارسال و پس از چک کردن و صحت اطلاعات ورودی، رکورد مربوطه را نمایش می دهد.
توجه: در تمام مثال ها، تمامی فیلد ها از نوع رشته ای در نظر گرفته شدند.

رویداد کلیک دکمه در صفحه اول:

```
protected void Button1_Click(object sender, EventArgs e)
{
    Session.Add("id", TextBox1.Text);
    Session.Add("pass", TextBox2.Text);
    Response.Redirect("~/Pages/LoginVerify.aspx");
}
```

و در صفحه مقصد (در اینجا با نام و آدرس Pages/LoginVerify.aspx):

در بخش فضای نام

```
using System.Data.OleDb;
```

و در بخش لود صفحه

```

protected void Page_Load(object sender, EventArgs e)
{
    string ConnectionString, QueryString;
    ConnectionString = "provider=microsoft.jet.oledb.4.0;data source="+Server.MapPath("~/DB\\bookshopDB.mdb");
    using (OleDbConnection Connection = new OleDbConnection(ConnectionString))
    {
        Connection.Open();
        QueryString = "select * from users where username='" + Session["id"] + "' and password='" + Session["pass"] + "'";
        OleDbCommand Command = new OleDbCommand(QueryString, Connection);
        OleDbDataReader Reader = Command.ExecuteReader();
        if (Reader.HasRows)
        {
            Reader.Read();
            TextBox1.Text = Reader["username"].ToString();
            TextBox2.Text = Reader["password"].ToString();
            TextBox3.Text = Reader["fname"].ToString();
            TextBox4.Text = Reader["email"].ToString();
            TextBox5.Text = Reader["zipcode"].ToString();
        }
        else
            Label1.Text = "نام کاربری یا رمز عبور اشتباه است";
        Reader.Close();
        Connection.Close();
    }
}

```

مثال) برنامه ای که نام کاربری را خوانده و در صورت موجود بودن آن، رکورد مربوطه را حذف می نماید.
 کد زیر در رویداد دکمه مورد نظر قرار می گیرد.

```
protected void Button1_Click(object sender, EventArgs e)
{
    string ConnectionString, QueryString;
    ConnectionString = "provider=microsoft.jet.oledb.4.0;data source=" + Server.MapPath("~/DB\\bookshopDB.mdb");
    using (OleDbConnection Connection = new OleDbConnection(ConnectionString))
    {
        Connection.Open();
        QueryString = "delete from users where username='" + TextBox1.Text + "'";
        OleDbCommand Command = new OleDbCommand(QueryString, Connection);
        Command.ExecuteNonQuery();
        Label1.Text = "رکورد حذف شد";
        Connection.Close();
    }
}
```

مثال) برنامه ای که شناسه یکتا، نام، تلفن و آدرس را از ورودی خوانده و آنرا در جدول address پایگاه داده درج می نماید.

```
protected void Button3_Click(object sender, EventArgs e)
{
    string ConnectionString, QueryString;
    ConnectionString = "provider=microsoft.jet.oledb.4.0;data source=" + Server.MapPath("~/DB\\mine.mdb");
    using (OleDbConnection Connection = new OleDbConnection(ConnectionString))
    {
        Connection.Open();
        QueryString = "insert into address(id,name,tel,adr) values('" + TextBox1.Text + "','" + TextBox2.Text + "','" + TextBox3.Text + "','" + TextBox4.Text + "')";
        OleDbCommand Command = new OleDbCommand(QueryString, Connection);
        Command.ExecuteNonQuery();
        Label1.Text = "رکورد درج شد";
        Connection.Close();
    }
}
```

مثال) برنامه ای که شناسه ای را از ورودی خوانده و رکورد مربوطه را از جدول address ویرایش می کند.

```
protected void Button4_Click(object sender, EventArgs e)
```

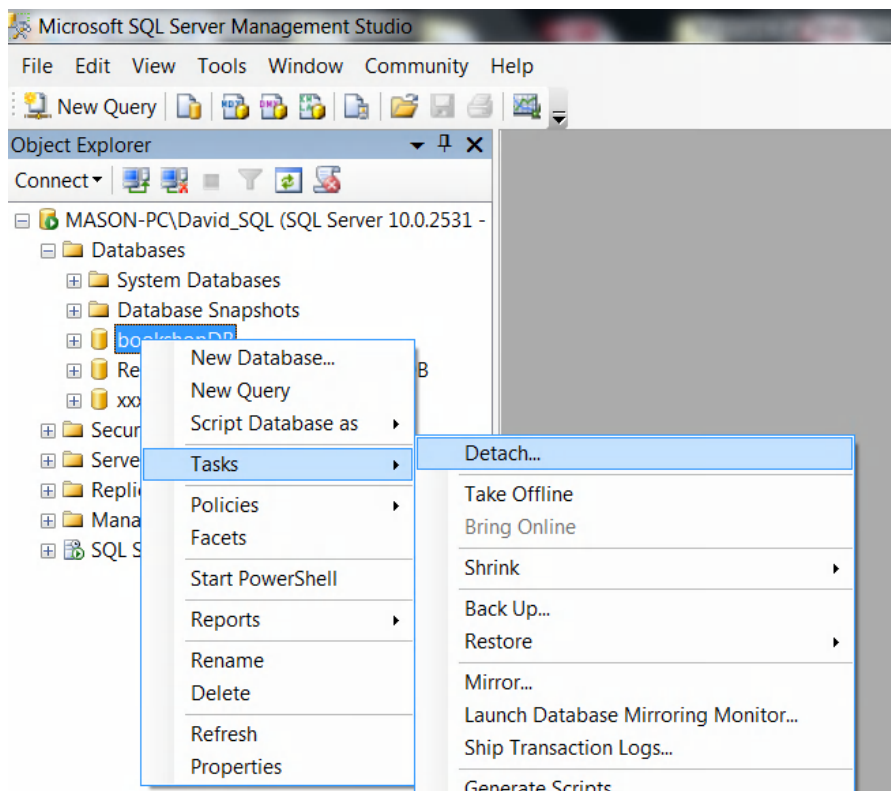
```
{
    string ConnectionString, QueryString;
    ConnectionString = "provider=microsoft.jet.oledb.4.0;data source=" + Server.MapPath("~/DB/mine.mdb");
    using (OleDbConnection Connection = new OleDbConnection(ConnectionString))
    {
        Connection.Open();
        QueryString = "update address set name='" + TextBox2.Text + "',tel='" + TextBox3.Text + "',adr='" + TextBox4.Text + "' where id='" + TextBox1.Text + "'";
        OleDbCommand Command = new OleDbCommand(QueryString, Connection);
        Command.ExecuteNonQuery();
        Label1.Text = "رکورد بروز شد";
        Connection.Close();
    }
}
```

روش انتقال و نسخه برداری از فایل پایگاه داده SQL

برای انتقال فایل دیتابیس در ابتدا باید اتصال DBMS را از فایل مورد نظر قطع کنید و سپس نسبت به کپی فایل اقدام نمایید. در غیر اینصورت DBMS از هر گونه کپی برداری، حذف، برش و... فایل ها جلوگیری به عمل می آورد.

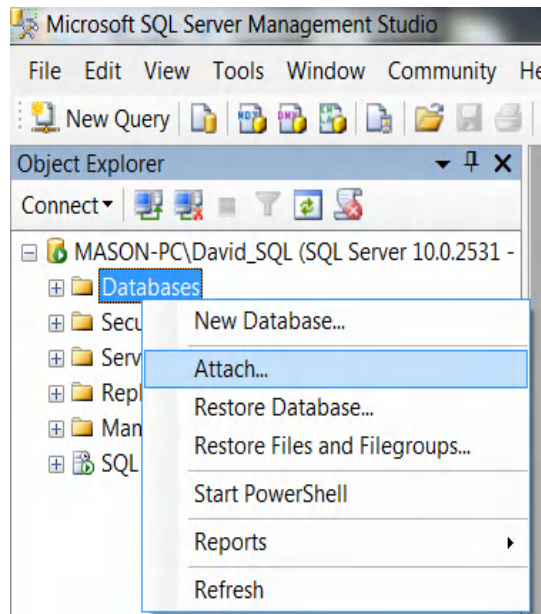
مراحل:

۱- در سیستم مبدا: در محیط اسکيوال بر روی دیتابیس مورد نظر عمل Detach را انجام دهید (طبق تصویر)



سپس فایل مورد نظر را نسخه برداری و در سیستم مقصد در مسیر دلخواه قرار دهید.

۲- در سیستم مقصد:



سپس در پنجره باز شده بر روی دکمه Add کلیک کنید و فایل دلخواه را آدرس دهی کنید.

سطح دسترسی در SQL

معمولاً بطور پیشفرض سطح دسترسی در اسکیوال چگونه ای است که کاربر قادر به انجام دستی بسیاری اعمال و تنظیمات نظیر درج و اصلاح و حذف و طراحی پرسوجو و... نمی باشد، برای افزایش سطح دسترسی، مراحل زیر اجرا شود. (در محیط اسکیوال)

۱- در مسیر

Tools\Options\Designers\Table and Database Designers

تیک گزینه Prevent saving changes that require table re-creation برداشته شود.

۲- بر روی دیتا بیس مورد نظر کلیک راست و در مسیر

Properties\Files\Owner\Browse

تیک گزینه sa زده شود.

ارتباط با پایگاه داده از طریق Wizard

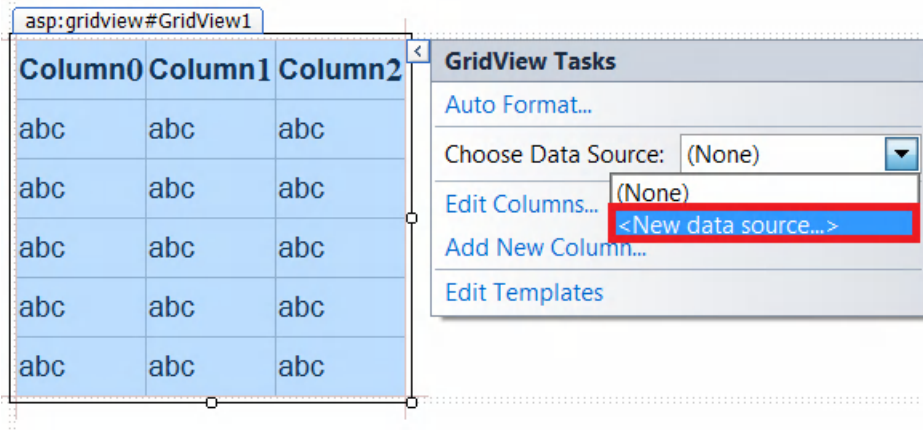
روش ارتباط دادن کلیه عناصر ویزاردی با بانک اطلاعاتی تقریباً مشابه هستند. یکی از کاربری ترین کنترل ها برای نمایش محتویات بانک اطلاعاتی GridView است.

(الف)

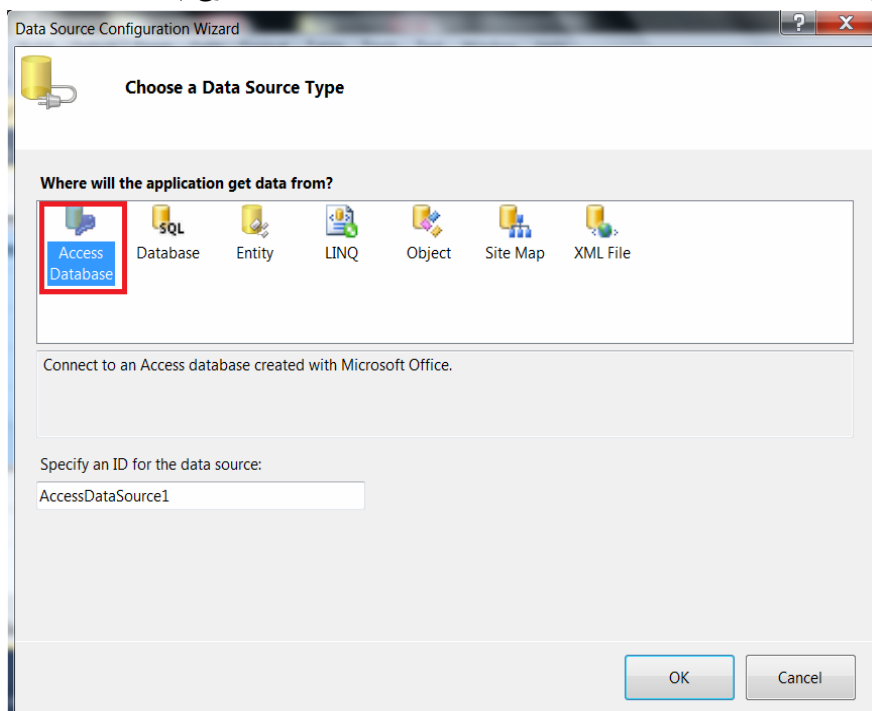
ارتباط پایگاه داده Access و GridView

ابتدا کنترل GridView را در فرم قرار داده و سپس مراحل زیر طی شود.

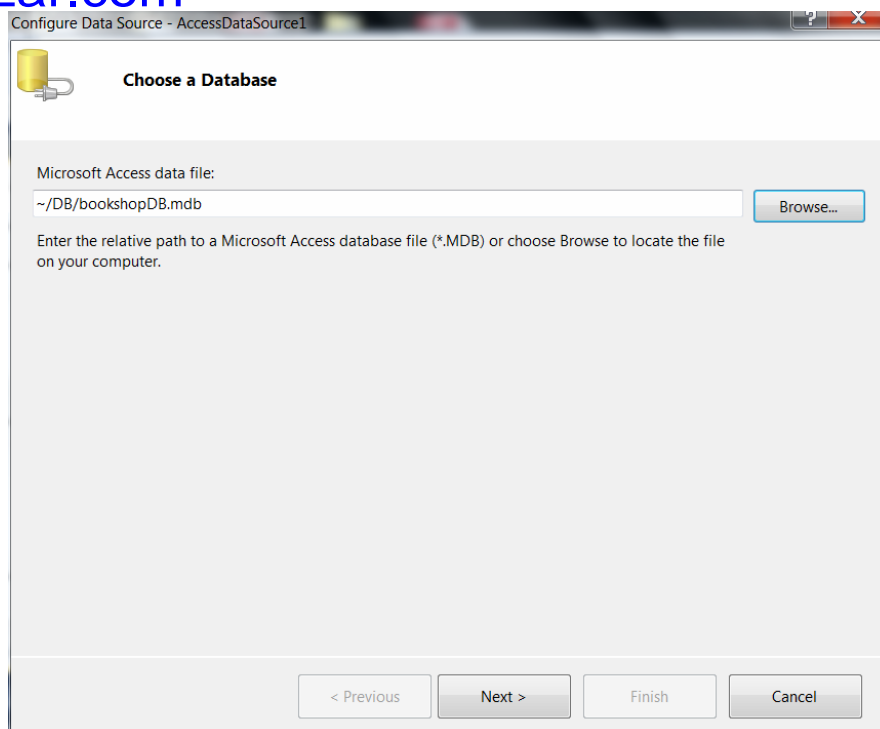
۱-



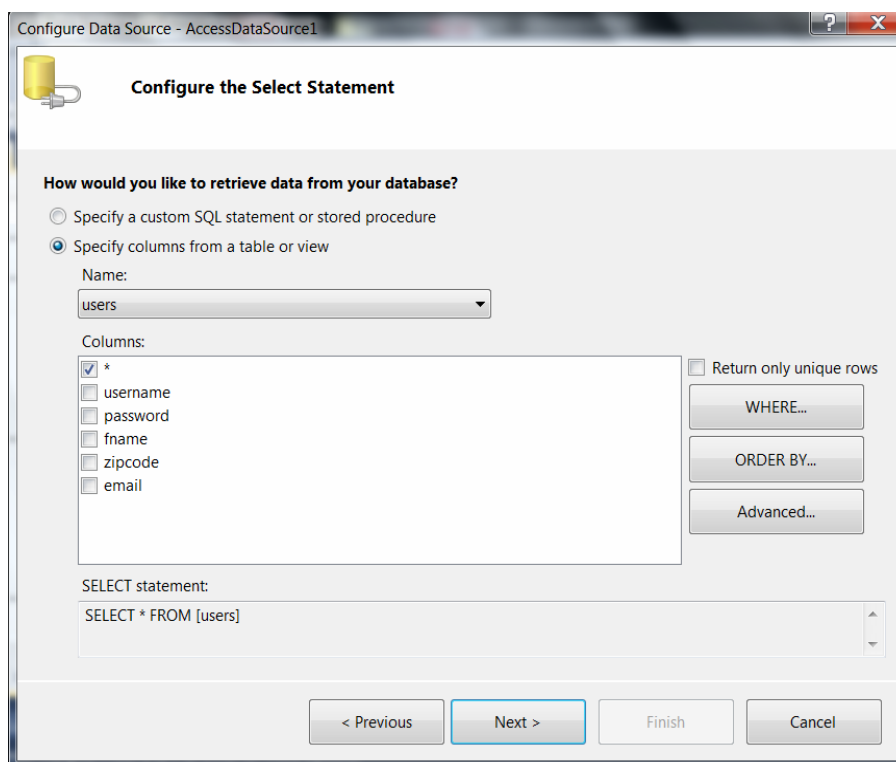
۲- در پنجره باز شده (Data Source Configuration Wizard) نوع پایگاه داده انتخاب میشود.



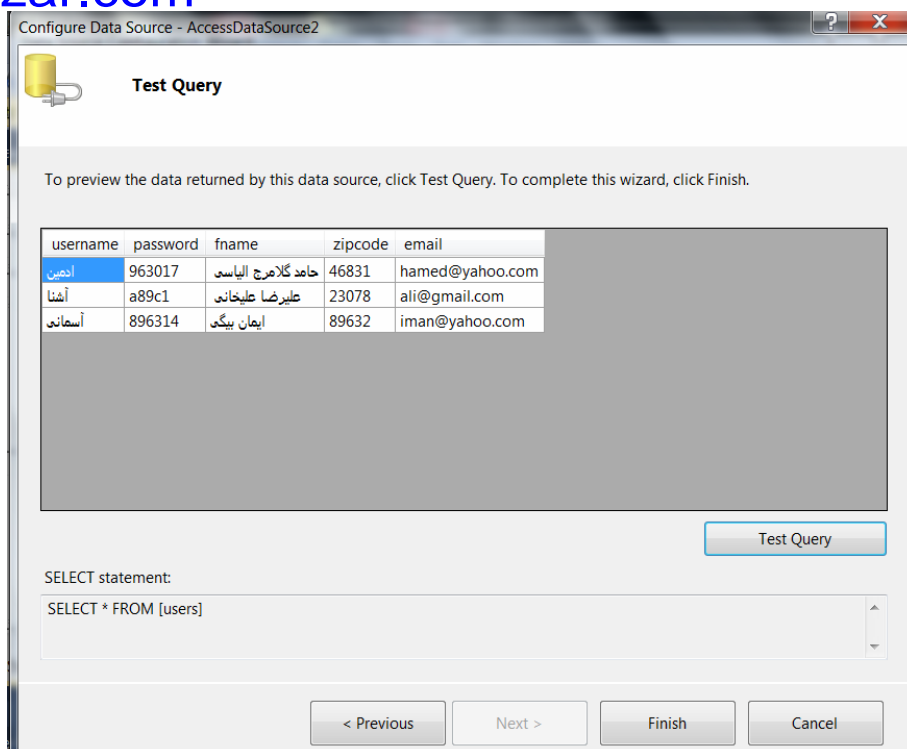
۳- در بخش بعدی (Choose a Database) و در قسمت Microsoft Access data file مسیر فایل پایگاه داده انتخاب شود.



۴- در قسمت (Configure the Select Statement) باید Query دلخواه خود را که مایل به نمایش آن هستید، ایجاد کنید .



۵- در بخش پایانی با کلیک بر روی دکمه Test Query می توانید از Query طراحی شده، تست پیش نمایش بگیرید.



(ب)

ارتباط پایگاه داده SQL و GridView

مراحل ۱ و ۲ و ۴ و ۵ کاملاً شبیه به بخش ارتباط Access و GridView است.

۳- در پنجره Choose Your Data Connection بر روی دکمه New Connection کلیک کنید تا پنجره Add Connection باز شود، در این پنجره:

Server Name: نام کامپیوتر قرار می گیرد، با توجه به متغیر بودن نام در کامپیوتر های مختلف توصیه می شود برای جلوگیری از بروز خطا، بجای انتخاب نام بصورت دستی، از . یا (local) استفاده شود، که به معنی نام سیستم جاری است.

Select or enter a data base name: نام دیتا بیس را از لیست موجود انتخاب کنید. با کلیک بر روی دکمه **Test Connection** می توانید اتصال را تست کنید.

اعمال فیلتر بر روی داده های در GridView

مثال) برنامه ای که رکورد فیلد هایی (در اینجا فیلد نام کاربری) که با کلمه یا جمله دلخواه (در اینجا متن ورودی از TextBox3) آغاز می شوند را در GridView نمایش می دهد. در رویداد کلیک دکمه مورد نظر کد زیر نوشته شود.

```
protected void Button2_Click(object sender, EventArgs e)
```

```
{
    string ConnectionString, QueryString;
    ConnectionString = "provider=microsoft.jet.oledb.4.0;data source=" + Server.MapPath("~/DB\\bookshopDB.mdb");
    using (OleDbConnection connection = new OleDbConnection(ConnectionString))
    {
        connection.Open();
        QueryString = "select * from users where username like'" + TextBox3.Text + "%'";
        OleDbCommand dbcom = new OleDbCommand(QueryString, connection);
        OleDbDataReader reader = dbcom.ExecuteReader();
        if (reader.HasRows)
        {
            GridView1.DataSource = reader;
            GridView1.DataBind();
        }
        else
        {
            Label1.Text = "سابقه موجود نیست";
        }
        connection.Close();
    }
}
```

WebConfig

یک فایل XML است که پیکره بندی وب سایت در آنجا معین می شود. تاکنون در هر بخش از برنامه که رشته اتصال (ConnectionString) مورد نیاز بود باید آنرا تکرار می کردیم، حال با قرار دادن رشته اتصال در WebConfig دیگر به تکرار مداوم رشته اتصال نیست بلکه در صورت نیاز آنرا فراخوانی می کنیم، بنابراین کد نویسی کاهش میابد. بطور پیشفرض در WebConfig دو تگ غیر فعال بصورت زیر موجود است که باید آنها را پاک کرد.

```
<appSettings/>
```

```
<connectionStrings/>
```

حال برای فعالسازی این تگ ها و قرار دادن رشته اتصال، آنها را بصورت زیر می نویسیم.

```
<appSettings>
```

```
  <add key="constr" value="provider=microsoft.jet.oledb.4.0;data source=D:\\book_shop\\database\\bookshop.mdb"/>
```

```
</appSettings>
```

```
<connectionStrings>
```

```
</connectionStrings>
```

Syntax عبارت بدینصورت است:

```
<add key="رشته اتصال" value="نام متغییر رشته اتصال"/>
```

سپس در بخش هایی از برنامه که نیاز به فراخوانی رشته اتصال موجود در WebConfig داریم به صورت زیر عمل می شود:

در ابتدای صفحه، فضای نام زیر اضافه شود:

```
using System.Configuration;
```

(توجه: همانطور که در بخش پایگاه داده گفته شد، کماکان وجود فضای نام `using System.Data.OleDb` برای کار با پایگاه داده به قوت خود باقیست).

سپس برای قرار دادن رشته اتصال، آنرا بدینصورت صدا می زنیم.

```
ConfigurationManager.AppSettings["نام متغییر رشته اتصال اضافه شده در وبکانفیگ"]
```

در مثال می شود:

```
string ConnectionString= ConfigurationManager.AppSettings["constr"];
```

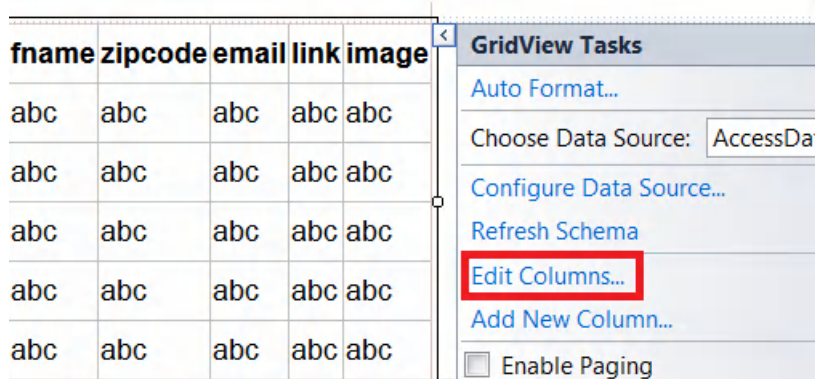
نکته: تعریف چند `ConnectionString` در `WebConfig` امکانپذیر است به عبارت دیگر می توان چند رشته اتصال تعریف کرد و هر گاه نیاز بود آنها را فراخوانی کرد.

درج تصویر و پیوند در GridView

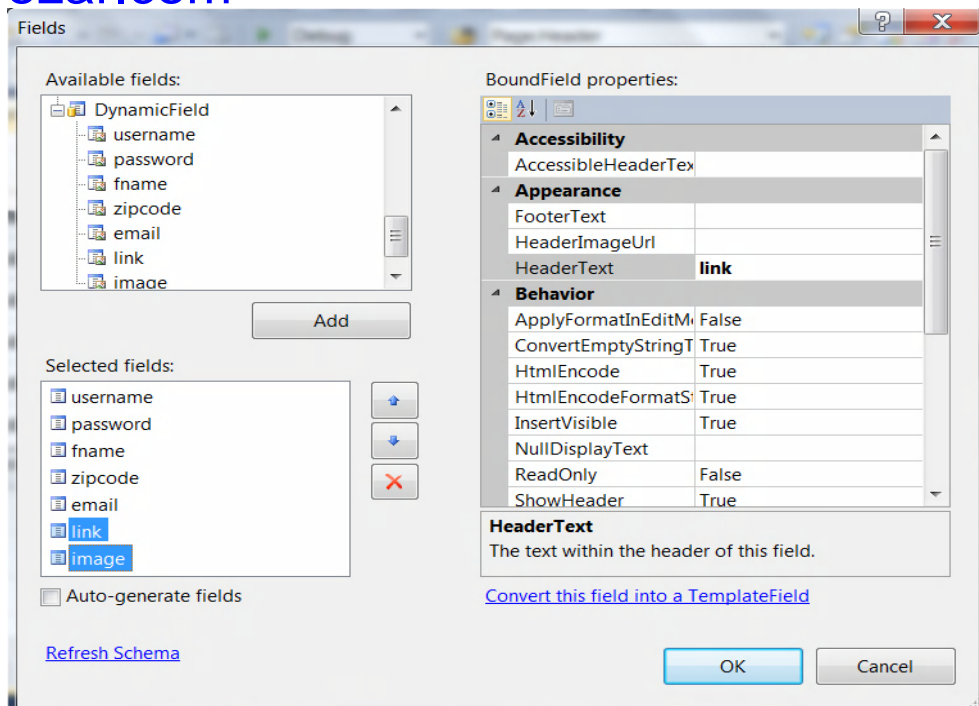
مثال) درج تصویر و پیوند در GridView

در فایل پایگاه داده، فیلدهای `image` و `link` را اضافه می کنیم که اولی آدرس فایل تصویر و دومی لینک صفحه یا سایتی دلخواه است.

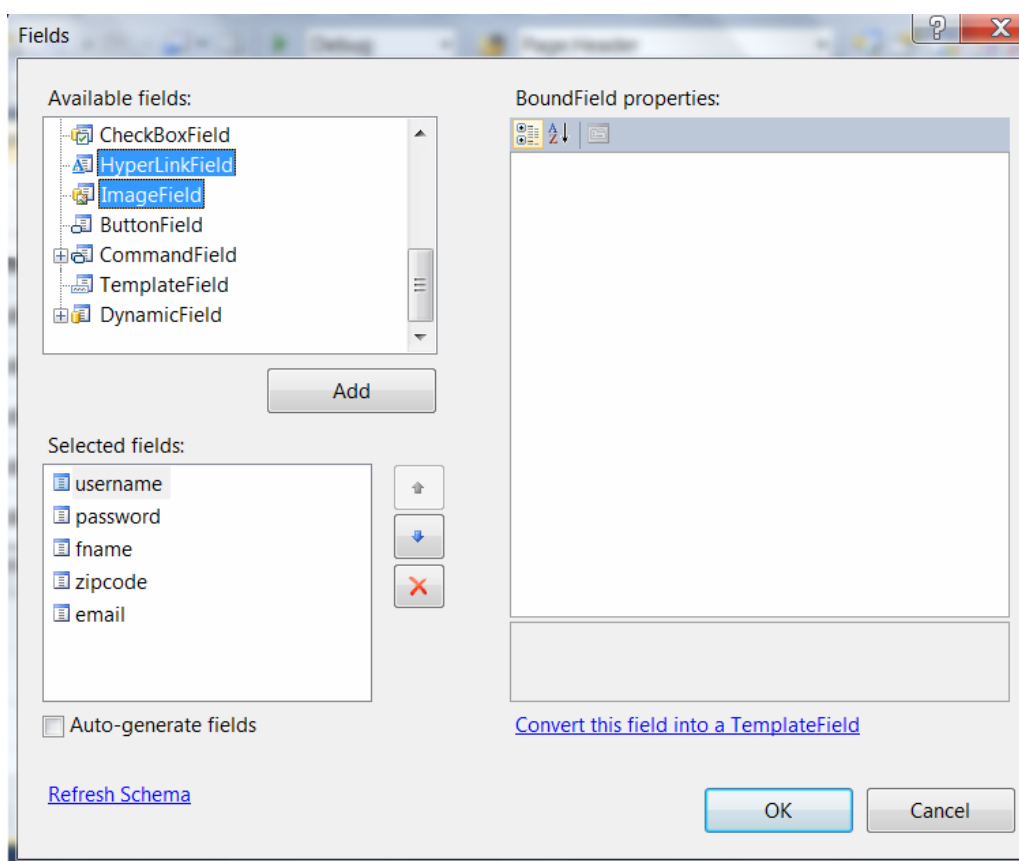
یک کنترل `GridView` در صفحه مورد نظر قرار دهید و آنرا به فایل پایگاه داده اتصال دهید. روی گزینه `Edit Columns` کلیک کنید تا وارد پنجره `Fields` شوید.



در این پنجره فیلدهای `image` و `link` را از لیست `Selected fields` انتخاب کرده و آنها را حذف کنید.



سپس از لیست Available Fields گزینه های HyperLinkField و ImageField را انتخاب و با کلیک بر روی گزینه Add آنها را به لیست Selected fields اضافه کنید.



از لیست Selected fields فیلد ImageField را انتخاب کنید و از بخش ImageField properties خاصیت DataAlternateTextField و DataImageUrlField (به ترتیب صفت آدرس فایل عکس و

صفت توضیح عکس) را برابر image قرار دهید و در خاصیت HeaderText متنی دلخواه برای عنوان وارد کنید (مثلا Picture).

از لیست Selected fields فیلد HyperLinkField را انتخاب کنید و از بخش ImageField properties خاصیت DataNavigateUrlFields و DataTextField (به ترتیب صفت آدرس سایت و صفت توضیح) را برابر link قرار دهید و در خاصیت HeaderText متنی دلخواه برای عنوان وارد کنید (مثلا personal page).

نمونه خروجی GridView قبل از اجرای مثال)

username	password	fname	zipcode	email
ادمین	963017	حامد گلنرج الیاسی	46831	hamed@hotmail.com
آشنا	a89c1	علیرضا علیخانی	23078	ali@yahoo.com
آسمانی	896314	ایمان بیگی	89632	iman@gmail.com

نمونه خروجی GridView پس از اجرای مثال)

username	password	fname	zipcode	email	picture	personal page
ادمین	963017	حامد گلنرج الیاسی	46831	hamed@hotmail.com		www.msn.com
آشنا	a89c1	علیرضا علیخانی	23078	ali@yahoo.com		www.yahoo.com
آسمانی	896314	ایمان بیگی	89632	iman@gmail.com		www.google.com

نکته: آدرس در فیلد های پایگاه داده می بایست به فرم استاندارد ASP.net باشند مثلا
c:\images\pic.jpg یا ~\picfolder\photo.bmp

AJAX

مجموعه فن آوری هایی است که به کمک آن لازم نیست در هر رخداد صفحه دوباره بارگذاری شود و مناسب جهت وب سایت های تعاملی است.

کنترل ScriptManager : به منزله موتور جاواست و در هر صفحه ای که از اجاکس استفاده می شود باید از آن استفاده کرد.

کنترل UpdatePanel : هر کنترلی که توسط اجاکس در حال تغییر و بروزرسانی است باید در درون این پنل قرار گیرد.

(مثال) برنامه نمایش استان و شهر های آن ها بکمک اجاکس.

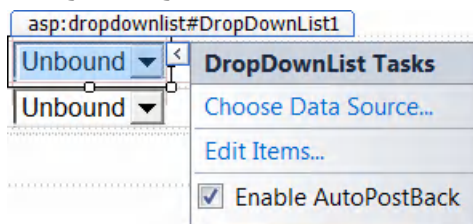
ابتدا جدولی با نام situation و با سه فیلد code ، state ، city طراحی می کنیم، که محتوای آنها بترتیب زیر هستند : الف) کد (کلید است) ، ب) نام استان ، ج) نام شهری که در آن استان قرار دارد اطلاعات موجود در جدول نمونه ما به این صورت است.

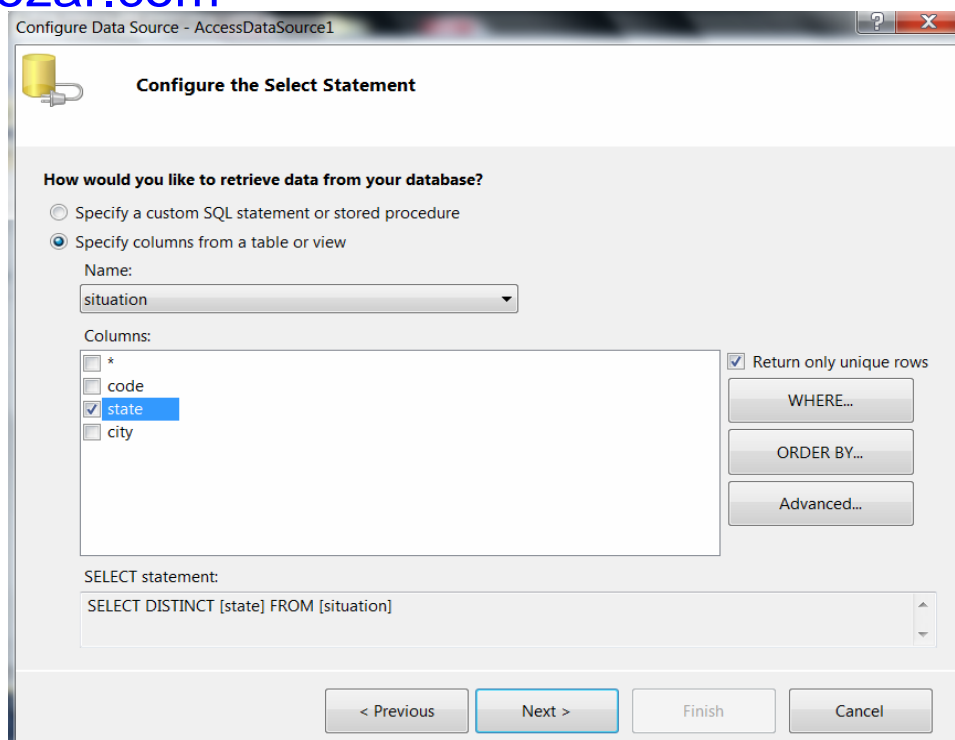
code	state	city	Add New Field
0	مازندران	نشتارود	
1	تهران	ری	
2	تهران	تهران	
3	مازندران	رامسر	
4	مازندران	تنکابن	
5	گیلان	رشت	
6	گیلان	چابکسر	
7	گیلان	انزلی	
8	گیلان	رودبار	
9	تهران	دماوند	
*	b		

حال در بخش طراحی فرم یک کنترل ScriptManager و یک UpdatePanel و دو DropDownList قرار می دهیم (یکی از DropDownList ها درون UpdatePanel قرار می گیرد، به این دلیل که با انتخاب نام استان از DropDownList اول، نام شهرها در DropDownList دوم بروز رسانی و محتوای آن تغییر می کند).

در DropDownList1:

تیک گزینه Enable AutoPostBack زده شود سپس به فیلد state از جدول situation اتصال داده شود، در هنگام اتصال، در پنجره Configure the select statement برای جلوگیری از نمایش تکراری نام استانها، گزینه Return only unique rows را تیک بزنید





سپس در رویداد SelectedIndexChanged آن کد زیر را قرار دهید.

```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    string ConnectionString, QueryString;
    ConnectionString = ConfigurationManager.AppSettings["constr"];
    QueryString = "select city from situation where state='" + DropDownList1.SelectedValue + "'";
    using (OleDbConnection Connection = new OleDbConnection(ConnectionString))
    {
        Connection.Open();
        OleDbCommand Command = new OleDbCommand(QueryString, Connection);
        OleDbDataReader reader = Command.ExecuteReader();

        DropDownList2.Items.Clear();
        while (reader.Read())
            DropDownList2.Items.Add(reader["city"].ToString());

        reader.Close();
        Connection.Close();
    }
}
```

توجه: مطابق روال همیشگی بدلیل استفاده از پایگاه داده و webconfig ، فضای نام های زیر باید ضمیمه شوند

```
using System.Data.OleDb;
using System.Configuration;
```

توجه: بجای استفاده از حرف ی از ترکیب کلید های shift+x استفاده شود.

آپلود و دانلود فایل

برای آپلود از کنترل FileUpload استفاده می شود.
 برای آپلود فضای نام using System.IO نیاز است.
 برای آپلود سه بخش اساسی سرور مورد توجه است: آدرس فایل، نام فایل، محل ذخیره سازی بر روی سرور.
 مثال) برنامه ای که آدرس فایلی را دریافت و پس از تایید حجم مجاز و نام غیر تکراری، آنرا آپلود می کند در غیر اینصورت پیغام مناسب صادر می شود.

```
protected void Button1_Click(object sender, EventArgs e)
{
    string savepath = Server.MapPath("~/upload");
    string savefile;
    if (FileUpload1.HasFile)
    {
        savefile = Path.Combine(savepath, FileUpload1.FileName);
        if (FileUpload1.PostedFile.ContentLength <= 150 * 1024)
        {
            if (!File.Exists(savefile))
            {
                FileUpload1.SaveAs(savefile);
                Label1.Text = "فایل ارسال شد";
            }
            else
                Label1.Text = "نام فایل تکراری است";
        }
        else
            Label1.Text = "فایل دارای حجم بیش از 150 کیلو بایت است";
    }
    else
        Label1.Text = "فایلی جهت ارسال انتخاب نشده است";
}
```

دانلود فایل: دارای روش های بسیار متنوعی است و با استفاده از کنترل های HyperLink، LinkButton، GridView و... قابل اجراست.
 کفایت در این کنترل ها لینک مقصد را به آدرس فایل مورد نظر خود هدایت کنید.

XML

انواع زبان ها:

۱- رویه ای: زبان هایی با ساختار کامل که امنیت، ظاهر، رویدادها، تبادل اطلاعات و... در آنها قابل کنترل است.

۲- توصیفی: زبان هایی مثل html و sql که فاقد ساختار هایی همچون حلقه، شرط و... هستند.

HTML دارای تگ های طولانی و سنگین است به همین علت در برخی سیستم عامل ها مخصوصا محیط هایی با منابع سخت افزاری کم (نظیر موبایل) قابل اجرا نیست.
ساختار ساده و کلی HTML بدینصورت است

```
<html>
<head>
بخش غیر محتوایی نظیر تعیین نام فونت و سایز فونت و... (که دیده نمی شوند)
<head/>
<body>
بدنه یا همان محتوا (که دیده می شود)
<body/>
</html>
```

XML دارای تگ های شکسته شده و سبک است بطوری که در تمام سیستم عامل ها با تمامی سخت افزار ها قابل اجراست.(Free Platform)

تگ ها در XML:

```
<تگ>
.....
.....
.....
</تگ>
```

توضیحات در XML:

```
<!--
توضیحات
-->
```

برای کار با فایل های XML به فضاهای نام زیر نیاز است.

```
using System.IO;
using System.Xml;
using System.Xml.Schema;
using System.Data;
```

مراحل کار با فایل XML: الف) ایجاد فایل ب) ایجاد ریشه ج) درج جدول د) درج رکورد

مثال) برنامه ای که با کلیک بر روی دکمه اول یک فایل XML تولید و با کلیک بر دکمه دوم، اطلاعات درون فایل تولید شده را در GridView نمایش میدهد. (توضیح هر بخش در درون سورس داده شده است) ابتدا فضاهای نام زیر ضمیمه شوند

```
using System.IO;
using System.Xml;
using System.Xml.Schema;
using System.Data;
```

در رویداد کلیک دکمه اول

```

protected void Button1_Click(object sender, EventArgs e)
{
    //ایجاد فایل XML
    FileStream fs = new FileStream("c:\\booklist.xml", FileMode.Create);
    XmlTextWriter w = new XmlTextWriter(fs, null);

    //ایجاد ریشه
    w.WriteStartDocument();
    w.WriteStartElement("booklist");
    w.WriteComment("XML تست ایجاد فایل");

    //درج اولین رکورد
    w.WriteStartElement("product"); //نام رکورد
    w.WriteAttributeString("id", "", "1"); //فیلد اول
    w.WriteAttributeString("name", "", "مهندسی نرم افزار"); //فیلد دوم
    w.WriteAttributeString("price", "", "4500"); //فیلد سوم
    w.WriteAttributeString("state", "", "6"); //فیلد چهارم
    w.WriteEndElement();

    //درج دومین رکورد
    w.WriteStartElement("product"); //نام رکورد
    w.WriteAttributeString("id", "", "2"); //فیلد اول
    w.WriteAttributeString("name", "", "طراحی وب"); //فیلد دوم
    w.WriteAttributeString("price", "", "5000"); //فیلد سوم
    w.WriteAttributeString("state", "", "1"); //فیلد چهارم
    w.WriteEndElement();

    //درج سومین رکورد
    w.WriteStartElement("product"); //نام رکورد
    w.WriteAttributeString("id", "", "3"); //فیلد اول
    w.WriteAttributeString("name", "", "پایگاه داده"); //فیلد دوم
    w.WriteAttributeString("price", "", "5200"); //فیلد سوم
    w.WriteAttributeString("state", "", "10"); //فیلد چهارم
    w.WriteEndElement();

    //بستن ریشه فایل
    w.WriteEndElement();
    w.WriteEndDocument();
    w.Close();
    Label1.Text = "فایل با موفقیت ایجاد شد";
}

```

در رویداد کلیک دکمه دوم

```
protected void Button2_Click(object sender, EventArgs e)
{
    DataSet ds = new DataSet();
    ds.ReadXml("c:\\booklist.xml");
    GridView1.DataSource = ds;
    GridView1.DataBind();
}
```

خروجی:

id	name	price	state
1	مهندسی نرم افزار	4500	6
2	طراحی وب	5000	1
3	پایگاه داده	5200	10

فایل با موفقیت ایجاد شد

ایجاد

نمایش

برای درک بهتر XML، محتویات فایل XML تولید شده را با سورس مثال بالا مقایسه نمایید.
فایل XML تولید شده (booklist.xml) به این صورت است:

```
<?xml version="1.0" ?>
<booklist>
  <!-- XML تست ایجاد فایل -->
  <product id="1" name="مهندسی نرم افزار" price="4500" state="6" />
  <product id="2" name="طراحی وب" price="5000" state="1" />
  <product id="3" name="پایگاه داده" price="5200" state="10" />
</booklist>
```