

همان طور که می‌دانید کاربرد پذیری در خیلی از پروژه‌ها حرف اول رو می‌زند و کاربر دوست دارد کارهایی که انجام می‌دهد خیلی راحت و با استفاده از موس باشد. یکی از کارهایی که در اکثر پروژه‌ها نیاز است، چیدمان ترتیب رکوردها است. ما می‌خواهیم در این پست ترتیبی اتخاذ کنیم که کاربر بتواند رکوردها را به هر ترتیبی که دوست دارد نمایش دهد.

ایجاد یک مورد جدید

عنوان	توضیحات	فعال	
مدل کلاسیک	طرح های قدیمی شیرآلات	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
مدل اهرمی	این مدل از طرح های جدید و نوین بهره می برد.	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
سینک شاور	توضیحاتی در مورد سینک های شاور	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
دسته ها و بوسته ها	توضیحاتی در مورد دسته ها و بوسته ها	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
علم های دوش	توضیحاتی در مورد علم های دوش	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
سایر محصولات		<input checked="" type="checkbox"/>	ویرایش مشاهده حذف

از توضیحاتی که [قبلا](#) دادم مشخص است که این کار احتمالا در ASP.NET WebForm کار سختی نیست ولی این کار باید در MVC از ابتدا طراحی شود.

طرح سوال : یک سری رکورد از یک Table داریم که می‌خواهیم به ترتیب وارد شدن رکوردها نباشد و ترتیبی که ما می‌خواهیم نمایش داده شود.

پاسخ کوتاه : خب باید ابتدا یک فیلد (برای اولویت بندی) به Table اضافه کنیم بعد اون فیلد رو بنا به ترتیبی که دوست داریم رکوردها نمایش داده شود پر کنیم (Sort می‌کنیم) و در آخر هم هنگام نمایش در View رکوردها را بر اساس این فیلد نمایش می‌دهیم.

(این پست هم در ادامه [پست قبلی](#) در همان پروژه است و از همان Table ها استفاده شده است)

اضافه کردن فیلد :

ابتدا یک فیلد به Table مورد نظر اضافه می‌کنیم. من اسم این فیلد رو Priority گذاشتم. Table من چنین وضعیتی دارد.

AMIRHOSSEIN-PC.Khazar - dbo.Types			
	Column Name	Data Type	Allow Nulls
	Id	int	<input type="checkbox"/>
	Title	nvarchar(200)	<input checked="" type="checkbox"/>
	Description	text	<input checked="" type="checkbox"/>
	ImagePath	varchar(500)	<input checked="" type="checkbox"/>
	Priority	int	<input checked="" type="checkbox"/>
	IsActive	bit	<input type="checkbox"/>
	CreatedOn	datetime	<input type="checkbox"/>
	CreatedBy	uniqueidentifier	<input type="checkbox"/>
	ModifiedOn	datetime	<input checked="" type="checkbox"/>
	ModifiedBy	uniqueidentifier	<input checked="" type="checkbox"/>
	IsDeleted	bit	<input type="checkbox"/>
			<input type="checkbox"/>

افزودن فایل‌های jQuery UI :

در این مرحله شما نیاز دارید فایل‌های مورد نیاز برای Sort کردن رکوردها را اضافه کنید. شما می‌توانید فقط فایل‌های مربوط به Sortable را به صفحه خودتان اضافه کنید و یا مثل من فایل‌هایی که حاوی تمام قسمت‌های jQuery UI هست را اضافه کنید.

من برای این کار از Section استفاده کردم ، ابتدا در Head فایل Layout دو Section تعریف کردم برای CSS و JavaScript . و فایل‌های مربوط به Sort کردن را در صفحه ای که باید عمل Sort انجام بشود در این Section ها قرار دادم.

فایل Layout

```
<head>
<meta charset="utf-8" />
@RenderSection("meta", false)
<title>@ViewBag.Title</title>
<link href="@Url.Content("~/Content/~Site.css")" rel="stylesheet" type="text/css" />
<link href="@Url.Content("~/Content/redactor/css/redactor.css")" rel="stylesheet" type="text/css" />
<link href="@Url.Content("~/Content/css/bootstrap-rtl.min.css")" rel="stylesheet" type="text/css" />
@RenderSection("css", false)
<script src="@Url.Content("~/Scripts/jquery-1.8.2.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/modernizr-1.7.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Content/js/bootstrap-rtl.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Content/redactor/redactor.min.js")" type="text/javascript"></script>
<script src="@Url.Content("~/Scripts/jquery.unobtrusive-ajax.min.js")" type="text/javascript"></script>
@RenderSection("js", false)
</head>
```

```
@model IEnumerable<KharazCo.Models.Type>
@{
    ViewBag.Title = "Index";
    Layout = "~/Areas/Administrator/Views/Shared/_Layout.cshtml";
}
@section css
{<link href="@Url.Content("~/Content/themes/base/jquery-ui.css")" rel="stylesheet" type="text/css" />}
@section js
{
    <script src="@Url.Content("~/Scripts/jquery-ui-1.9.0.min.js")" type="text/javascript"></script>
}
```

در آخر فایل Index.html به اینصورت شده است:

```
<h2>
    نوع ها
</h2>
<p>
    @Html.ActionLink("ایجاد یک مورد جدید", "Create", null, new { @class = "btn btn-info" })
</p>
<table>
    <thead>
        <tr>
            <th>
                عنوان
            </th>
            <th>
                توضیحات
            </th>
            <th>
                فعال
            </th>
            <th>
            </th>
            <th>
            </th>
        </tr>
    </thead>
    <tbody>
        @foreach (var item in Model.OrderBy(m => m.Priority))
        {
            <tr id="@item.Id">
                <td>
                    @Html.DisplayFor(modelItem => item.Title)
                </td>
                <td>
                    @(new HtmlString(item.Description))
                </td>
                <td>
                    @Html.DisplayFor(modelItem => item.IsActive)
                </td>
                <td>
                    @Html.ActionLink("ویرایش", "Edit", new { id = item.Id }, new { @class = "btnEdit
label label-warning" })
                    @Html.ActionLink("مشاهده", "Details", new { id = item.Id }, new { @class =
"btnDetails label label-info" })
                    @Html.ActionLink("حذف", "Delete", new { id = item.Id }, new { @class = "btnDelete
label label-important" })
                </td>
            </tr>
        }
    </tbody>
</table>
```

** توجه داشته باشید که من به هر tr یک id اختصاص داده ام که این مقدار id همان مقدار فیلد Id همان رکورد هست ، ما برای مرتب کردن به این Id نیاز داریم (خط 25).

افزودن کدهای کلاینت :

حالا باید کدی بنویسیم که دو کار را برای ما انجام دهد : اول حالت Sort پذیری را به سطرهای Table بدهد و دوم اینکه هنگامی که ترتیب سطرهای تغییر کرد ما را با خبر کند:

```
<script type="text/javascript">
    $(function () {
        $("table tbody").sortable({
            helper: fixHelper,
            update: function (event, ui) {
                jQuery.ajax('@Url.Action("Sort", "Type", new { area = "Administrator" })', {
                    data: { s: $(this).sortable('toArray').toString() }
                });
            }
        }).disableSelection();
    });
    var fixHelper = function (e, ui) {
        ui.children().each(function () {
            $(this).width($(this).width());
        });
        return ui;
    };
</script>
```

توضیح کد :

در این کد ما حالت ترتیب پذیری را به Table می دهیم و هنگامی که عمل Update در Table انجام شد تابع مربوطه اجرا می شود. ما در این تابع، ترتیب جدید سطرها را می گیریم (** به کمک مقدار Id که به هر سطر دادیم ، این مقدار Id برابر بود با Id خود رکورد در Database) و به کمک jQuery.ajax به تابع Sort از کنترلر Type در منطقه (Administrator) area ارسال می کنیم و در آنجا ادامه کار را انجام می دهیم.

تابع fixHelper هم به ما کمک می کند که هنگامی که سطرها از جای خود جدا می شوند ، دارای عرض یکسانی باشند و عرض آنها تغییری نکند.

افزودن کد Server :

حالا باید تابع Sort که مقادیر را به آن ارسال کردیم بنویسیم. من این تابع را بر اساس مقداری که از کلاینت ارسال می شود اینگونه طراحی کردم .

```
public EmptyResult Sort(string s)
{
    if (s != null)
    {
```

```
public EmptyResult Sort(string s)
{
    if (s != null)
    {
        var ids = new List<int>();
        foreach (var item in s.Split(','))
```

```

    {
        ids.Add(int.Parse(item));
    }
    int intpriority = 0;
    foreach (var item in ids)
    {
        intpriority++;
        db.Types.Single(m => m.Id == item).Priority = intpriority;
    }
    db.SaveChanges();
}
return new EmptyResult();
}

```

در ابتدا مقادیر Id که از کلاینت به صورت String ارسال شده است را می‌گیریم و بعد به همان ترتیب ارسال در لیستی از int قرار می‌دهیم ids.

سپس به اضافی هر رکورد Type مقدار اولویت را به فیلدی که برای همین مورد اضافه کردیم Priority اختصاص می‌دهیم. و در آخر هم تغییرات را ذخیره می‌کنیم. (خود کد کاملاً واضح است و نیازی به توضیح بیشتر نیست)

حالا باید هنگامی که لیست Type ها نمایش داده می‌شود به ترتیب (OrderBy) فیلد Priority نمایش داده شود پس تابع Index را اینطور تغییر می‌دهیم.

```

public ActionResult Index()
{
    return View(db.Types.Where(m => m.IsDeleted == false).OrderBy(m => m.Priority));
}

```

این هم خروجی کار من:

نوع ها

ایجاد یک مورد جدید

عنوان	توضیحات	فعال	
مدل اهرمی	این مدل از طرح های جدید و نوین بهره می برد.	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
مدل کلاسیک	طرح های قدیمی شیرآلات	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
سینک شاور	توضیحاتی در مورد سینک های شاور	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
دسته ها و بوسته ها	توضیحاتی در مورد دسته ها و بوسته ها	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
علم های دوش	توضیحاتی در مورد علم های دوش	<input checked="" type="checkbox"/>	ویرایش مشاهده حذف
سایر محصولات		<input checked="" type="checkbox"/>	ویرایش مشاهده حذف

این عکس مربوط به است به قسمت مدیریت پروژه [شیرآلات مرجان خزر](#) .

نظرات خوانندگان

نویسنده:

سعید

تاریخ:

۲۱:۱۲ ۱۳۹۱/۰۹/۰۸

« این کار احتمالا در ASP.NET WebForm کار سختی نیست »

اتفاقا کار ساده‌ای نیست و همین مراحل باید طی شود. ضمن اینکه کار با jquery ajax در آنجا به این یک دستی نیست. نیاز است در code behind فرم، متد وب سرویس مانندی به صورت استاتیک تعریف شود (که خودش سبب می‌شود تا دسترسی به اعتبار سنجی توکار مبتنی بر فرم‌ها محدود شود) یا اینکه از یک هندلر مجزا بجای یک اکشن متد کمک گرفته شود ... خلاصه خیلی داستان دارد.

نویسنده:

امیرحسین مرجانی

تاریخ:

۲۱:۱۶ ۱۳۹۱/۰۹/۰۸

من خودم این کار رو در webform انجام ندادم و خیلی هم با سختی هاش آشنایی ندارم. ولی فکر کردم شاید با update panel بشه این کار رو راحت انجام داد. خب نظرتون راجب به راه حل من چیه؟

راه حل خوبی ارائه دادم؟ متن قابل فهم بود؟

نویسنده:

سعید

تاریخ:

۲۱:۲۰ ۱۳۹۱/۰۹/۰۸

بسیار عالی.

فقط در مورد update panel ... تنها کاری که انجام می‌دهد کپسوله کردن ارسال مقادیر به سرور است به صورت ajax سازگار با کنترل‌های دارای view state. بیشتر از این کاری انجام نمی‌دهد. مابقی آن اگر یک سری کنترل در toolkit آن موجود باشد برای این کارها یا خیر. این toolkit هم محدود است و آنچنان به روز نمی‌شود.

نویسنده:

امیرحسین مرجانی

تاریخ:

۲۱:۲۳ ۱۳۹۱/۰۹/۰۸

ممنونم از پاسختون

از اینکه سوییچ کردم روی MVC خوشحالم (:

البته یکی از دلایلی هم که این کار رو کردم همین راحتی استفاده Ajax بود.

نویسنده:

امیر آشنا

تاریخ:

۲۲:۴ ۱۳۹۱/۰۹/۰۸

سلام

مطلب مفیدی بود ولی می‌تونستید بیشتر توضیح بدید.

برای ما افراد مبتدی درک بعضی قسمت‌ها کمی مشکل هست.

ولی باز هم ممنونم

نویسنده:

امیرحسین مرجانی

تاریخ: ۲۲:۷ ۱۳۹۱/۰۹/۰۸

سلام

بفرمایید کدوم قسمت‌ها تا بیشتر توضیح بدم.

نویسنده: وحید نصیری
تاریخ: ۲۲:۲۱ ۱۳۹۱/۰۹/۰۸

پیشنیازهای مطلب جاری:

- در مورد RenderSection ([^](#))
- مقدمه‌ای بر jQuery Ajax در MVC ([^](#))
- db.SaveChanges : کل مباحث Entity framework سایت ([^](#))
- EmptyResult و کلا خروجی‌های اکشن متدها: ([^](#))
- علت استفاده از Sort@ Url.Action در حین آدرس دهی: ([^](#))

نویسنده: امیرحسین مرجانی
تاریخ: ۲۲:۲۳ ۱۳۹۱/۰۹/۰۸

واقعا تشکر می‌کنم بابت این لیستی که ارائه کردید.

بحث تکمیل شد

به نظر شما باید بیشتر به جزییات اهمیت داده می‌شد؟

نویسنده: وحید نصیری
تاریخ: ۲۲:۴۱ ۱۳۹۱/۰۹/۰۸

ضرورتی نداره. چون واقعا به اندازه لیستی که عنوان شد نیاز به پیشنیاز درک این مطالب هست و فرصت تکرار آن‌ها نیست. این مطالب جدید، یک سری مطالب تکمیلی هستند نه مطالب پایه و از صفر.

نویسنده: محسن موسوی
تاریخ: ۰:۲۴ ۱۳۹۱/۰۹/۰۹

با استفاده از [Web API Controller](#) تقریبا به همین راحتی امکانپذیره. تغییر زیادی هم نیاز نداره.

نویسنده: امیرحسین مرجانی
تاریخ: ۰:۳۷ ۱۳۹۱/۰۹/۰۹

ممنونم آقای موسوی

قابل توجه آقا سعید (احتمالا این مطلب برای شما مفیده)

نویسنده: سعید
تاریخ: ۰:۵۲ ۱۳۹۱/۰۹/۰۹

- نیاز به vs2012 داره.

- نمیشه یک فرم رو strongly typed تعریف کرد مثل مثال بالا:

```
@model IEnumerable<KhazarCo.Models.Type>
```

نویسنده: محسن موسوی
تاریخ: ۱۰:۲۱ ۱۳۹۱/۰۹/۰۹

بحث در مورد UI نبود. بحث در مورد استفاده کردن بود.
در ضمن در Web Form ها نیز از امکاناتی شبیه به MVC می توان بهره برد. (^)

نویسنده: سعید
تاریخ: ۱۳۹۱/۰۹/۰۹ ۱۰:۴۶

فقط mvc هست که امکان استفاده از چند view engine را با هم دارد.

صفحه ای که لینک دادید مربوط است به [web pages](#) asp.net و نه web forms. این web pages برای کار با webmatrix طراحی شده.

نویسنده: محسن موسوی
تاریخ: ۱۳۹۱/۰۹/۰۹ ۱۱:۴۰

- استفاده از این قابلیت در [Web Page](#) ها توسط VS
 - استفاده از این قابلیت در [Web Form](#) ها توسط VS
- هر دو مورد توسط VS استفاده شده است.

نویسنده: سعید
تاریخ: ۱۳۹۱/۰۹/۰۹ ۱۱:۴۵

```
<iframe src="/twitter" frameborder="0" height="400" width="270" scrolling="no"></iframe>
```

فلسفه و کاربرد web pages متفاوت است. در مورد وب فرم ها هم نمی تونید از razor استفاده کنید چون در یک فایل نمی شود از دو موتور view استفاده کرد. موتور view وب فرم ها، نامش همین view engine web forms است و قابل تعویض هم نیست (برخلاف MVC). در مقاله ای که لینک دادید، داره از یک *iframe* استفاده می کنه برای الحاق razor.

به علاوه در مورد Web API که کلا مطلب جدیدی نیست. نسخه ساده شده WCF است.

نویسنده: محسن موسوی
تاریخ: ۱۳۹۱/۰۹/۰۹ ۱۲:۲۷

دوست عزیز
بحث ما در مورد توانایی انجام موارد فوق الذکر بود. نه اینکه حالا چون ...
به توانایی های MVC شکی نیست. مسئله اصلی اینه که آیا پست جاری را میتوان به راحتی با WebPage و یا WebForm انجام داد؟!
چه از طریق قابلیت های ASP.NET Web Form و ASP.NET Web Page راحتی.
استفاده از Razor چه از طریق JQuery و یا IFrame (نظر قبلی) و یا [به طور مستقل](#) و یا روش های دیگر در ASP.NET Web Form و ASP.NET Web Page
و استفاده معمولی با توانایی های DataBind
یا بطور کامل از [Razor View Engine](#) در Web Page
در نهایت کار سختی نیست.

نویسنده: سعید
تاریخ: ۱۳۹۱/۰۹/۰۹ ۱۲:۴۳

razor بی جهت اینجا طرح شد. من در مورد strongly typed view سطری رو نوشتم. این نوع view ها مستقل از نوع view

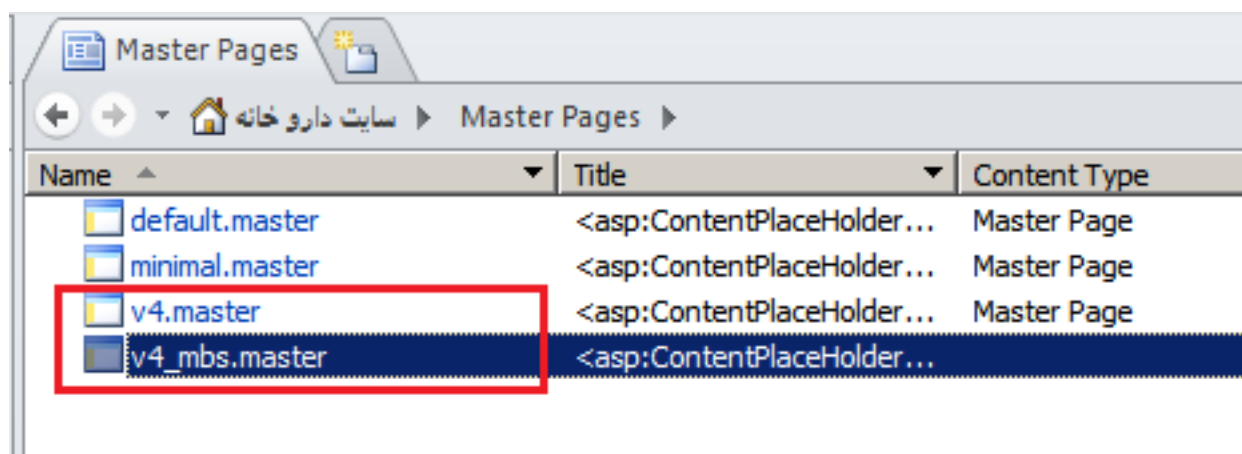
engine هستند. در mvc حتی با موتور web forms هم میشه یک چنین view های تحت نظر کامپایلری رو با پشتیبانی کامل از intellisense داشت.

یکی از نیازهای مشتریان هنگام استفاده از سایت‌های تحت شیرپوینت ، عدم نمایش نوار مدیریتی بالای صفحه یا همان Ribbon برای کاربران ناشناس است .

شاید بتوان گفت که مزیت این راهکار نسبت به دیگر راه کارها ، تصحیح نمایش Scroll Bar مرورگر است که ممکن است در برخی روش‌ها با مشکل مواجه شود. دلیل این امر هم این است که شیرپوینت برای افزودن Ribbon به بالای صفحه Vertical Scroll Bar را از صفحه حذف می‌کند و سپس Scroll Bar سفارشی خود را به صفحه طوری اضافه می‌کند تا نوار Ribbon همیشه در بالاترین نقطه از صفحه بماند

همچنین در این روش از بارگذاری نوار هنگام بالا آمدن سایت نیز البته برای کاربران ناشناس جلوگیری می‌شود و اما روش :

سایت خود را با SharePoint Designer باز کرده و از Master Page یک کپی تهیه کنید و آن را به عنوان پیش فرض سایت تعیین کنید



سپس تگ زیر را در صفحه بیابید :

```
<div id="s4-ribbonrow">
```

و تگ زیر را به ابتدای آن (قبل از تگ) اضافه کنید :

```
<Sharepoint:SPSecurityTrimmedControl runat="server" Permissions="AddDelPrivateWebParts">
```

مانند تصویر زیر :

```
<SharePoint:DelegateControl runat="server" ControlId="GlobalNavigation"/>
<!-- Start -->
<SharePoint:SPSecurityTrimmedControl runat="server" Permissions="AddDelPrivateWebParts">

<div id="s4-ribbonrow" class="s4-pr s4-ribbonrowhidetitle">
  <div id="s4-ribboncont">
    <SharePoint:SPRibbon
      runat="server"
      PlaceholderElementId="RibbonContainer"
      CssFile="">
      <SharePoint:SPRibbonPeripheralContent
```

و تگ پایانی آن :

```
    <ContentTemplate>
      <WebPartPages:WebPartAdder ID="WebPartAdder" runat="server" />
    </ContentTemplate>
  <Triggers>
    <asp:PostBackTrigger ControlID="WebPartAdder" />
  </Triggers>
</asp:UpdatePanel>
</div>
</div>
<!-- END -->

</SharePoint:SPSecurityTrimmedControl>

<div id="s4-workspace">
```

نکته مهم در استفاده از این تگ ، ویژگی Permissions آن است که باید با دقت و بسته به نیاز شما تعریف شود :

برخی از این موارد عبارتند از :

.EmptyMask – Has no permissions on the Web site. Not available through the user interface

.ViewListItems – View items in lists, documents in document libraries, and view Web discussion comments

.AddListItems – Add items to lists, add documents to document libraries, and add Web discussion comments

EditListItems – Edit items in lists, edit documents in document libraries, edit Web discussion comments in documents, and customize Web Part Pages in document libraries

DeleteListItems – Delete items from a list, documents from a document library, and Web discussion comments in documents

.ApproveItems – Approve a minor version of a list item or document

.OpenItems – View the source of documents with server-side file handlers

.ViewVersions – View past versions of a list item or document

.DeleteVersions – Delete past versions of a list item or document

.CancelCheckout – Discard or check in a document which is checked out to another user

.ManagePersonalViews – Create, change, and delete personal views of lists

.ManageLists – Create and delete lists, add or remove columns in a list, and add or remove public views of a list

.ViewFormPages – View forms, views, and application pages, and enumerate lists

.Open – Allow users to open a Web site, list, or folder to access items inside that container

.ViewPages – View pages in a Web site

AddAndCustomizePages – Add, change, or delete HTML pages or Web Part Pages, and edit the Web site using a SharePoint Foundation-compatible editor

.ApplyThemeAndBorder – Apply a theme or borders to the entire Web site

.ApplyStyleSheets – Apply a style sheet (.css file) to the Web site

.ViewUsageData – View reports on Web site usage

.CreateSSCSite – Create a Web site using Self-Service Site Creation

.ManageSubwebs – Create subsites such as team sites, Meeting Workspace sites, and Document Workspace sites

.CreateGroups – Create a group of users that can be used anywhere within the site collection

ManagePermissions – Create and change permission levels on the Web site and assign permissions to users and groups

BrowseDirectories – Enumerate files and folders in a Web site using Microsoft Office SharePoint Designer 2007 and WebDAV interfaces

.BrowseUserInfo – View information about users of the Web site

.AddDelPrivateWebParts – Add or remove personal Web Parts on a Web Part Page

.UpdatePersonalWebParts – Update Web Parts to display personalized information

ManageWeb – Grant the ability to perform all administration tasks for the Web site as well as manage content. Activate, deactivate, or edit properties of Web site scoped Features through the object model or through the user interface (UI). When granted on the root Web site of a site collection, activate, deactivate, or edit properties of site collection scoped Features through the object model. To browse to the Site Collection Features page and activate or deactivate site collection scoped Features through the UI, you must be a site collection administrator.

UseClientIntegration – Use features that launch client applications; otherwise, users must work on documents locally and upload changes.

UseRemoteAPIs – Use SOAP, WebDAV, or Microsoft Office SharePoint Designer 2007 interfaces to access the Web site.

ManageAlerts – Manage alerts for all users of the Web site.

CreateAlerts – Create e-mail alerts.

EditMyUserInfo – Allows a user to change his or her user information, such as adding a picture.

EnumeratePermissions – Enumerate permissions on the Web site, list, folder, document, or list item.

FullMask – Has all permissions on the Web site. Not available through the user interface.

حال خارج از تگ‌های SPSecurityTrimmedControl در ابتدا یا انتها ، باید تگ login را مانند زیر به آن اضافه کرد .

```
<!-- END -->
```

```
</SharePoint:SPSecurityTrimmedControl>
```

```
<asp:LoginView id="LoginView" runat="server">
  <AnonymousTemplate>
    <style type="text/css">
      body { overflow-y: scroll !important; overflow-x: hidden; }
      body #s4-workspace { overflow-x: hidden; overflow-y: auto !important; }
    </style>
    <!--[if lte IE 7]>
    <style type="text/css">
      html { overflow: auto !important; overflow-x: hidden; }
      body { overflow: auto !important; }
    </style>
    <![endif]-->
  </AnonymousTemplate>
</asp:LoginView>
```

و تمام :



[موفق باشید](#)