



دانشگاه شهاب دانش
(غیر دولتی - غیر انتفاعی)

دانشگاه شهاب دانش

دانشکده مهندسی برق و کامپیوتر و مهندسی پزشکی

پایان نامه کارشناسی

در رشته مهندسی کامپیوتر گرایش تکنولوژی نرم افزار

عنوان

سایت خبری Asp.net MVC

نگارش

علی عباسی

استاد راهنما

مهدی حاجی میری

بهمن ۱۳۹۷



فرم شماره ۴

بسمه تعالی

فرم صورتجلسه دفاع از پروژه نهایی
(درج در پایان نامه)



این قسمت توسط استاد راهنما تکمیل شود

نام و نام خانوادگی دانشجو:

شماره دانشجویی:

نام استاد راهنما:

نام استاد داور:

عنوان دقیق پروژه:

.....

نظر استاد راهنما و استاد داور

دفاع از پروژه‌ی فوق انجام شد و پایان نامه مورد قبول اینجانب می باشد.

نام و امضای استاد داور:

نام و امضای استاد راهنما:

تاریخ: ۱۳۹ / /

تاریخ: ۱۳۹ / /

نظر مدیر گروه آموزشی

دفاع از پروژه‌ی فوق انجام شد و مورد تایید اینجانب می باشد.

نام مدیر گروه آموزشی: امضا: تاریخ: ۱۳۹ / /

تأییدیه‌ی صحت و اصالت نتایج

باسمه تعالی

اینجانب علی عباسی به شماره دانشجویی ۹۵۴۱۱۱۵۲ دانشجوی رشته کامپیوتر مقطع تحصیلی کارشناسی ناپیوسته تأیید می‌نمایم که کلیه‌ی نتایج این پایان‌نامه/رساله حاصل کار اینجانب و بدون هرگونه دخل و تصرف است و موارد نسخه‌برداری‌شده از آثار دیگران را با ذکر کامل مشخصات منبع ذکر کرده‌ام. در صورت اثبات خلاف مندرجات فوق، به تشخیص دانشگاه مطابق با ضوابط و مقررات حاکم (قانون حمایت از حقوق مؤلفان و مصنفان و قانون ترجمه و تکثیر کتب و نشریات و آثار صوتی، ضوابط و مقررات آموزشی، پژوهشی و انضباطی ...) با اینجانب رفتار خواهد شد و حق هرگونه اعتراض درخصوص احقاق حقوق مکتسب و تشخیص و تعیین تخلف و مجازات را از خویش سلب می‌نمایم. در ضمن، مسئولیت هرگونه پاسخگویی به اشخاص اعم از حقیقی و حقوقی و مراجع ذیصلاح (اعم از اداری و قضایی) به عهده‌ی اینجانب خواهد بود و دانشگاه هیچ‌گونه مسئولیتی در این خصوص نخواهد داشت.

نام و نام خانوادگی:

امضا و تاریخ:

مجوز بهره‌برداری از پایان‌نامه

بهره‌برداری از این پایان‌نامه در چهارچوب مقررات کتابخانه و با توجه به محدودیتی که توسط استاد راهنما به شرح زیر تعیین می‌شود، بلامانع است:

- ☐ بهره‌برداری از این پایان‌نامه / رساله برای همگان بلامانع است.
- ☐ بهره‌برداری از این پایان‌نامه / رساله با اخذ مجوز از استاد راهنما، بلامانع است.
- ☐ بهره‌برداری از این پایان‌نامه / رساله تا تاریخ ممنوع است.

نام استاد یا اساتید راهنما:

تاریخ:

امضا:

چکیده

پروژه طراحی سایت خبری با هدف ارائه انواع اخبار متنی، تصویری، ویدیویی در انواع زمینه های سیاسی، اقتصادی، استانی، بین المللی، ورزشی، رسانه، دانش و فناوری و... طراحی شده به علاوه امکان حذف و اضافه و ویرایش انواع داده های درون سایت از قبیل گروه بندی خبر ها، برچسب های استفاده شده در سایت، مطالب خبری و رسانه های استفاده شده و... برای مدیر سایت در پنل مدیریت ارائه شده.

در پروژه سایت خبری سه عدد controller نوشته شده است:

- ❖ HomeController
- ❖ AdminController
- ❖ UserController

قسمت HomeController: مربوط به صفحه اصلی، ورود کاربران، ثبت نام کاربران و ورود مدیر میباشد.
قسمت AdminController: مربوط به صفحه خود مدیر، دیدن کاربران، دیدن خرید ها، دیدن مدیر های سایت، ویرایش مدیر ها و دیدن نظرات.

قسمت UserController: مربوط به صفحه خود کاربر، خرید و آپلود عکس، نمایش های خرید های قبلی، نمایش اطلاعات خود کاربر و گذاشتن نظر.

واژه های کلیدی:

سایت خبری، EntityFramWork و Controller، View، Model، Repository و...

صفحه

فهرست عناوین

۱	فصل اول مقدمه	۱
۴	فصل دوم ابزار های پیاده سازی	۲
۵	۲,۱ Asp.net mvc	۵
۵	2.1.1 :Model	۵
۵	2.1.2 : view	۵
۶	2.1.3 : Controller	۶
۷	2.1.4 مزیت های معماری mvc	۷
۷	2.1.5 فریم ورک هایی که بر پایه mvc هستند	۷
۸	۲,۲ معرفی نرم افزار Visual Studio	۸
۱۰	۲,۳ معرفی نرم افزار Microsoft SQL Server	۱۰
۱۰	۱,۳,۲ OLTP	۱۰
۱۱	۲,۳,۲ OLAP	۱۱
۱۲	۲,۳,۳ تاریخچه SQL Server	۱۲
۱۴	۲,۴ تکنولوژی LINQ	۱۴
۱۴	۵,۲ Ajax چیست	۱۴
۱۵	۳ فصل سوم نمودار ها و جداول	۱۵
۱۶	۳,۱ نمودار ER	۱۶
۱۷	۳,۲ نمودار کلاس	۱۷
۱۸	۳,۳ پایگاه داده	۱۸
۱۹	۴ فصل چهارم پیاده سازی نرم افزار	۱۹
۱۹	۱,۴ سناریو ها	۱۹
۲۳	۴,۲ UML Activity Diagram	۲۳
۲۶	۳,۴ الگوی Repository	۲۶
۲۶	۴,۴ Entity Framework چیست	۲۶
۳۰	۴,۵ اعتبار سنجی اطلاعات ورودی در فرم های ASP.NET MVC	۳۰
۳۱	۴,۶ Razor چیست	۳۱
۳۲	۴,۷ نحوه ایجاد controller	۳۲
۳۴	۸,۴ ActionResult چیست	۳۴

۳۵ ایجاد Layout page	۹,۴
۳۸ ActionVerbs متدهای	۴,۱۰
۳۹ [Http post] [Http get] متد	۱۱,۴
۴۰ view کردن برای ثبت نام	۴,۱۲
۴۵ htmlHelper متد	۱۳,۴
۴۸ فصل پنجم، جمع بندی و نتیجه گیری	۵
۵۰ منابع و مراجع	
۵۱ پیوست‌ها	

صفحه

فهرست اشکال

۷	شکل 1.3 visual studio
۹	شکل 2.3 sql server
۱۵	شکل 1.3 نمودار ER
۱۶	شکل 2.3 نمودار کلاس
۱۷	شکل 3.3 شرح جداول
۱۸	شکل ۱-۴ صفحه ثبت نام
۱۹	شکل ۴-۴ فلوچارت لاگین
۲۰	شکل ۵-۴ فلوچارت ثبت خبر
۲۶	شکل ۶-۴ ایجاد Entity FramWork مرحله ۱
۲۷	شکل ۷-۴ ایجاد Entity FramWork مرحله ۲
۲۷	شکل ۸-۴ ایجاد Entity FramWork مرحله ۳
۲۸	شکل ۹-۴ ایجاد Entity FramWork مرحله ۴
۳۰	شکل ۱۰-۴ اعتبارسنجی ثبت نام کاربران
۳۲	شکل ۱۱-۴ نحوه ایجاد contoroller مرحله ۱
۳۲	شکل ۱۲-۴ نحوه ایجاد contoroller مرحله ۲
۳۳	شکل ۱۳-۴ نحوه ایجاد contoroller مرحله ۳
۳۵	شکل ۱۴-۴ نحوه ایجاد Layout page مرحله ۱
۳۶	شکل ۱۵-۴ نحوه ایجاد Layout page مرحله ۲
۳۸	شکل ۱۶-۴ نحوه ایجاد Layout page مرحله ۳
۳۹	شکل ۱۷-۴ انواع ActionVerbs
۴۰	شکل ۱۸-۴ ایجاد View مرحله ۱
۴۱	شکل ۱۹-۴ ایجاد View مرحله ۲
۴۱	شکل ۲۰-۴ ایجاد View مرحله ۳
۴۵	شکل ۲۱-۴ ایجاد یک Html Helper با استفاده از Extension Method

صفحه

فهرست جداول

۱۹.....	جدول ۱-۴ سناریو ثبت نام
۲۱.....	جدول ۲-۴ سناریو ثبت خبر
۲۲.....	جدول ۳-۴ سناریو دیدن خبر های قبلی

۱

فصل اول

مقدمه

مقدمه

امروزه سیستم‌های اداری، سیستم‌های جهانی هستند که وظیفه اصلی آن‌ها ایجاد ارتباط و بهبود ارتباطات است. ارتباطات از لحاظ اطلاعات تجاری و خبری از اهمیت بسزایی برخوردار است و رمز بقای سازمان‌ها و تداوم فعالیت‌های آن‌ها است. مجهز شدن این سازمان‌ها به ابزارهای رقابتی عصر اطلاعات و ارتباطات؛ یعنی سیستم‌های اطلاعاتی و فناوری اطلاعات است. در این میان از دهه ۱۹۶۰ که جنبه‌های بیشتری از کاربرد و فعالیت‌های اداری و بازرگانی رو به گسترش گذاشت، وجود یک سیستم یکپارچه اداری مناسب که حجم عظیم اطلاعات، مکاتبات را در بر گیرد به وضوح احساس شد.

هدف از طراحی سایت خبری، گسترش سریع اخبار روز دنیا، آرشیو اخبار برای اطلاع‌رسانی به افراد می‌باشد که در این بین چگونگی نمایش اخبار، به‌روزترین اخبار، نحوه آرشیو اخبار از نکات مهم در طراحی سایت خبری بشمار می‌آید. همچنین نوع قالب، نحوه‌ی کد نویسی و همچنین با توجه به سرور آن سرعت لود صفحات یک سایت خبری را تحت تأثیر قرار می‌دهد که تأخیر زیاد در لود صفحه موجب می‌شود کارکنان تمایلی به خواندن سایت خبری نشان ندهند، که این از نکات مهم طراحی سایت خبری است. اما مهم‌ترین نکته در طراحی وبسایت خبری مسئله‌ی دیده شدن سایت از طریق مرورگرهای مختلف و نرخ ایندکس شدن می‌باشد. چراکه وبسایت خبری‌ای که بازدیدکننده‌ای نداشته باشد، یک طراحی سوخت شده و عملاً بدون کاربرد می‌باشد. به‌طور خلاصه وبسایت خبری‌ای موفق خواهد شد که به‌صورت استاندارد و روی اصول طراحی شده باشد و اصول سئو در آن رعایت شده باشد تا بتواند در نتایج گوگل رتبه‌ی بالایی را کسب کند و در صورت ثبت خبر جدید به‌سرعت توسط گوگل خواننده شود. طراحی سایت عصرمجازی در طول فعالیت حرفه‌ای خود توانسته است با داشتن سرور خارجی و ایرانی، متخصصین در زمینه برنامه‌نویسی و سئوکاران حرفه‌ای و طراحان قوی یکی از مدعیان در زمینه طراحی سایت خبری بشمار آید.

فصل دوم

ابزارهای پیاده سازی

۲.۱ سند مشخصات پیاده سازی

زبان برنامه نویسی : Asp.net mvc

محیط برنامه نویسی : Visual Studio 2015

پایگاه داده : SQL Server 2017

ابزار مدلسازی : Visual Studio 2015

استفاده از تکنولوژی LINQ در SQL

۲.۱.۱ Asp.net mvc

mvc مخفف کلمات Model View Controller هست . در واقع MVC بر روی معماری های چند لایه ای جهت جداسازی قسمت های مختلف برنامه است. هر یک از بخش های معماری MVC یعنی Model و View و controller را به شکل زیر تعریف می کنیم.

۲.۱.۲ Model

در mvc مدل ها همان کلاس های برنامه هستند که شبیه به دیتابیس هستند ولی می توانند متفاوت و بیشتر باشند. همچنین Model وظیفه تایید داده ها را برعهده دارد تا اطلاعات درست درون آنها قرار بگیرد و برای دیتابیس ارسال شود.

۲.۱.۳ view

این بخش اطلاعات را به کاربر نشان می دهد، ویو کاری به منطق برنامه ندارد و فقط یک model دریافت می کند و آن را به هر شکلی که نیاز باشد به کاربر نشان می دهد، این model هم توسط کنترلر ایجاد

شده است و برای view رسال شده است. نکته مهمی که در بخش View باید مد نظر داشت این است که این لایه مسئول کنترل صحت داده های وارد شده از طریق کاربر و همچنین مسئول صحت داده های نشان داده شده به کاربر نیست. در واقع این بخش با داده های خام کار میکند و مسولیت صحت داده با model است.

۲.۱.۴ Controller

این بخش همانگونه که از اسم آن مشخص است، یک بخش کنترل کننده است، ساده بگویم controller بیشتر مواقع دو جور کار می کند

۲.۱.۴.۱ Controller

اطلاعات را از دیتابیس در قالب model هایی می گیرد و بعد پردازش خود را روی آن انجام می دهد و اطلاعات را در قالب model جدیدی به view می فرستد تا ویو هر طور خواست این اطلاعات را به کاربر نشان دهد. مثل فرض کنید که ما برای صفحه پروفایل یک view برای موبایل داریم و یک view برای ویندوز، حال کنترلر بنا به تشخیص که کاربر با موبایل است یا نه یک view را انتخاب می کند تا برای کاربر ارسال شود و می بینید که یک اطلاعات مشخص در دو view متفاوت ظاهر می شود و view (ظاهر) از منطق برنامه جدا است.

۲.۱.۴.۲ controller

اطلاعات را از مدلی که از view برای او ارسال کرده است دریافت می کند و پس انجام اقداماتی آن را در دیتابیس ذخیره می کند و view دیگری را به کاربر می فرستد تا ببیند.

۲.۱.۵ مزیت های معماری mvc

- ۲.۱.۵.۱ استفاده دوباره از کد که در مهندسی نرم افزار بسیار مهم است، به دلیل جدا بودن این سه قسمت از هم این قابلیت به خوبی در این معماری قابل لمس است.
- ۲.۱.۵.۲ همان طور که در بالا توضیح داده شد ، به سادگی می توان چندین interface داشت. یعنی شما می توانید چندین view مختلف داشته باشید و به سادگی از آنها استفاده کنید.
- ۲.۱.۵.۳ کسی که ui کار است می تواند کاملاً به ظاهر و ui توجه کند و به مسائل منطق یا باگ های برنامه کاری نداشته باشد.
- ۲.۱.۵.۴ کسانی که کار منطق برنامه را پیدا می کنند می توانند فقط به کار خود توجه داشته باشند و به ظاهر صفحه ها کاری نداشته باشند.
- ۲.۱.۵.۵ بنابر این کار تیمی بروی یک پروژه به خوبی با mvc انجام می شود.
- ۲.۱.۵.۶ به دلیل جدا بودن view از model به سادگی می توان view اضافه کرد بدون این که model تغییری کند.

۲.۱.۶ فریم ورک هایی که بر پایه mvc هستند

اگر جواب mvc چیست را یافتید نوبت به آموزش mvc برای یکی از framework های محبوب است. در زیر فریم ورک هایی که بر پایه mvc هستند برای زبان های برنامه نویسی مختلف نام می برم و شما در این میان نام فریم ورک هایی را می بینید که بسیار این روز ها معروف شده اند و افراد قوی آنها را پیشنهاد می کنند که این خود دلیل دیگری برای اثبات خوب بودن mvc است.

Asp.net دیگر کنار رفته است و Asp.net mvc آمده است که در آن با معماری mvc کد C# می زنید و از توسعه وب لذت می برید. رسانه آموزشی لیموند، ورزش سه و ... از تکنولوژی asp mvc بهره برده اند

.Php به دلیل متن باز بودن فریم ورک های mvc زیادی دارد که در زیر نام می برم که در حال حاضر
لاراول محبوب ترین فریم ورک php است.

CodeIgniter

CakePHP

Zend

Yii

۲.۲ معرفی نرم افزار Visual Studio



شکل 1.3 visual studio

نرم افزار Visual Studio از پر کاربردترین نرم افزار ها در رشته مهندسی می باشد که کاربرد فراوانی در
تهیه پایان نامه های کارشناسی ارشد , مقالات علمی و پژوهش های مختلف دارد . در ادامه این آموزش به
معرفی نرم افزار ویژوال استودیو و کاربرد های آن پرداخته می شود . نرم افزار ویژوال استودیو

(Visual studio) شامل مجموعه‌ای از ابزارها و برنامه‌ها است که توسط شرکت مایکروسافت، برای ساخت سرویس‌ها و برنامه‌های کاربردی وب ASP.NET (فناوری طراحی و ایجاد صفحات وب)، سرویس‌های وب XML، برنامه‌های کنسول (console applications) برنامه‌های کاربردی دسکتاپ (Desktop) و موبایل (قابل اجرا بر روی دستگاه‌هایی که پلتفرم Microsoft Smartphone بر روی آنها پیاده‌سازی شده است) و همچنین برنامه‌های واسط کاربری گرافیکی (GUI)، ارائه شده است.

شرکت مایکروسافت برای سالها، به توسعه برنامه‌های اختصاصی مانند Visual C++ (ویژوال سی پلاس پلاس)، Visual J++، Visual InterDev و Visual Basic (ویژوال بیسیک) می‌پرداخت؛ اما در سال ۱۹۹۷ به منظور گرد هم‌آوری این محیط‌های برنامه‌نویسی، نرم‌افزار Visual Studio را در محیط توسعه یکپارچه (IDE) معرفی کرد. از آن زمان تاکنون نسخه‌های مختلفی از نرم‌افزار ویژوال استودیو منتشر شده است. ویژوال استودیو ۹۸ (به نام ویژوال استودیوی ۶ نیز شناخته می‌شود)، همزمان با انتشار ویژوال بیسیک ۶ و Visual Studio.NET 2002 نیز همزمان با اولین نسخه NET framework به بازار آمد. به‌روزرسانی نرم‌افزار Visual Studio ادامه پیدا کرد و نسخه‌های ۲۰۰۲، ۲۰۰۳، ... ۲۰۱۳ و ۲۰۱۵ آن نیز منتشر شد و اخیراً نسخه ۲۰۱۷ آن (Visual Studio 2017) نیز در دسترس قرار گرفته است. نرم‌افزار Visual Studio قابلیت اجرای برنامه‌ها را بر روی تمام پلتفرم‌های پشتیبانی شده توسط مایکروسافت ویندوز، Windows Phone، .NET Framework، Microsoft Silverlight و ... را دارد.

نرم‌افزار ویژوال استودیو هم‌اکنون در سه نسخه Community، Professional و Enterprise از طریق سایت: www.visualstudio.com عرضه می‌شود. نسخه Community رایگان بوده و بیشترین استفاده آن مربوط به تیم‌های برنامه‌نویسی کوچک و محققان دانشگاهی می‌باشد که می‌توانند در پایان نامه‌های کارشناسی ارشد و همچنین در پژوهش‌های خود از آن استفاده نمایند. در نسخه حرفه‌ای نرم‌افزار Visual Studio، امکاناتی از قبیل اشکال‌زدایی از راه دور و برنامه‌نویسی از طریق تلفن همراه اضافه شده است. نسخه Enterprise که جامع‌ترین نسخه نرم‌افزار Visual Studio است، برای توسعه‌دهندگان اصلی و برنامه‌نویسان خبره ارائه گردیده است. در نسخه Enterprise نرم‌افزار ویژوال استودیو امکانات منحصر به فردی از قبیل تست عملکرد و بارگذاری وب، مدیریت آزمایشگاه (Lab Management) و ... پیش‌بینی شده است. همان‌طور که در مقاله آموزشی بالا اشاره شد می‌توان نتیجه گرفت که نرم‌افزار Visual studio دارای کاربردهای فراوانی در رشته‌های مختلف مهندسی می‌باشد که پزشکین و فارغ

التحصیلان می توانند با استفاده از آن به ارایه پایان نامه ها و مقالات و پژوهش های خود بپردازند. جدول زیر امکانات موجود در هر سه نسخه نرم افزار ویژوال استودیو را نشان می دهد.

۲.۳ معرفی نرم افزار Microsoft SQL Server



شکل 2.3 sql server

یکی از سیستم های مدیریت بانک های اطلاعاتی رابطه ای (Relational) است که توسط شرکت مایکروسافت ارائه شده است. SQL Server از مدل سرویس دهنده - سرویس گیرنده (Client/Server) تبعیت می نماید. در این مدل، درخواست های (InQuery) سرویس گیرندگان برای سرویس دهنده ارسال و در سمت سرویس دهنده بررسی و آنالیز می گردند. در ادامه، پردازش های مورد نیاز بر روی اطلاعات ذخیره شده در بانک های اطلاعاتی انجام و در نهایت، نتایج برای سرویس گیرنده ارسال خواهد شد. MS SQL Server با استفاده از مجموعه عناصری (Components) که به صورت هدفمند اجراء می گردند، قادر به تامین نیازها و درخواست ها از مخازن داده (Data Storages) می باشد. مخازن داده در SQL Server به دو روش زیر مدیریت می گردند:

۲.۳.۱ OLTP

در مدل OLTP ، مخازن داده به صورت جداول رابطه ای که عموماً به جهت جلوگیری از تکرار و ناهمگونی اطلاعات به صورت هنجار (Normalize) درآمده اند، سازماندهی می شوند. این نوع از بانک های اطلاعاتی برای درج و تغییر سریع اطلاعات توسط چندین کاربر بطور همزمان مناسب می باشند.

OLAP ۲.۳.۲

در مدل OLAP مخازن داده جهت تجزیه و تحلیل و خلاصه سازی حجم زیادی از اطلاعات سازماندهی می شوند. مخازن داده و ارتباط بین اطلاعات در این مدل توسط SQL Server مدیریت می گردد. یکی از اهداف مهم سیستم های مدیریت بانک های اطلاعاتی، قابلیت رشد و توسعه (Scalability) است. MS SQL Server مجموعه ای از پتانسیل ها را به منظور تامین هدف فوق ارائه نموده است که به برخی از مهم ترین آنها اشاره می گردد :

قابلیت کار با بانک های اطلاعاتی حجیم (در حد ترابایت)

قابلیت دسترسی هزاران کاربر بطور همزمان به بانک اطلاعاتی

قابلیت خود سازگاری (Self Compatibility): با استفاده از ویژگی فوق، منابع مورد نیاز هر کاربر (نظیر حافظه، فضای دیسک و ...) به محض اتصال به سرور (Log in) به صورت اتوماتیک به وی تخصیص داده می شود و پس از Log off، منابع اختصاص یافته به منظور استفاده سایر کاربران آزاد می شوند.

قابلیت اعتماد و در دسترس بودن (Reliability): با استفاده از ویژگی فوق می توان بسیاری از فعالیت های مدیریتی را بدون توقف سرور انجام داد (نظیر پشتیبان گیری).

برخورداري از سطوح امنیتی بالا: بدین منظور اعتبارسنجی کاربران توسط SQL با اعتبارسنجی ویندوز تجمیع می گردد. در چنین مواردی، ضرورتی به تعریف کاربر در MS SQL نخواهد بود و اعتبارسنجی وی توسط ویندوز انجام خواهد شد.

پشتیبانی از حجم بالای حافظه فیزیکی در سرور (در نسخه ۲۰۰۰ تا ۶۴ گیگابایت و در نسخه ۲۰۰۵ و ۲۰۰۸ متناسب با حافظه ای که سیستم عامل از آن حمایت می نماید).

استفاده از چندین پردازنده به صورت موازی (در نسخه ۲۰۰۰ تا ۳۲ پردازنده همزمان و در نسخه ۲۰۰۵ و ۲۰۰۸ محدودیتی وجود ندارد)

پشتیبانی از لایه ها و سوکت های امنیتی نظیر SSL، خصوصاً جهت استفاده در وب ...

یکی دیگر از ویژگی های مهم سیستم های مدیریت بانک های اطلاعاتی، ایجاد تسهیلات لازم به منظور مدیریت بانک های اطلاعاتی است. بانک SQL Server با ارائه برنامه های جانبی نظیر Enterprise Manager، استفاده و مدیریت بانک های اطلاعاتی را آسان نموده است.

MS SQL Server بطور اتوماتیک در Active Directory ثبت می شود (Register)، بنابراین کاربران شبکه به راحتی می توانند آن را در Directory Active جستجو و در صورت نیاز به آن متصل شوند. همچنین، MS SQL Server توسط IIS پشتیبانی می گردد و مرورگرها با استفاده از پروتکل HTTP قابلیت استفاده از آن را خواهند داشت.

از جمله نکات مهم در خصوص MS SQL Server، اجرای آن به صورت یک سرویس است. بنابراین، در صورتی که کاربری به ماشینی که بانک MS SQL Server بر روی آن اجرا شده است، Log on نکرده باشد، همچنان سیستم در دسترس کاربران خواهد بود. علاوه بر این، می توان از سیستم مانیتورینگ ویندوز به منظور مانیتورینگ SQL Server استفاده نمود.

یکی از مهم ترین و شاخص ترین ویژگی های MS SQL Server که از نسخه ۲۰۰۰ در آن ایجاد شده است، امکان نصب چندین نسخه SQL بر روی یک ماشین می باشد (Multi Instance)، بطوریکه هر یک از نسخه ها فایل های باینری مخصوص به خود را داشته و بطور جداگانه مدیریت و راهبری می گردند، ولی تمام نسخه ها بطور همزمان اجرا می شوند (دقیقا" مشابه این است که چندین نسخه بر روی چندین کامپیوتر نصب شده باشد). ابزارهای همراه SQL نظیر Enterprise Manager SQL یک مرتبه نصب خواهند شد و در تمامی نسخه های SQL به صورت مشترک استفاده خواهند شد.

۲.۳.۳ تاریخچه SQL Server

شرکت IBM در دهه ۷۰ زبانی به نام SEQUEL را، خاص پرس و جو (query) از پایگاه های داده ابداع کرد. این واژه معادل زبان پرس و جوی ساخت یافته می باشد. به مرور بخش های جدیدی به این زبان اضافه شد تا جایی که دیگر نه تنها برای پرس و جو بلکه برای ساخت و مدیریت پایگاه داده و نکات ایمنی موتور پایگاه داده نیز بکار می رفت. شرکت IBM این زبان را به عموم عرضه کرد و پس از آن SQL نامیده

شد. موتورهای پایگاه داده امروزی ویرایش های متعددی از SQL را بکار می برند. ویرایش مورد استفاده SQL Server نیز Transact-SQL (T_SQL) نام دارد.

مایکروسافت در ابتدا SQL Server را با همکاری Sybase برای اجرا در محیط OS/2 طراحی کرد. هنگامی که مایکروسافت و IBM از هم جدا شدند، مایکروسافت کار را با سیستم عامل جدید و شبکه ای خود تحت عنوان سرویس دهنده پیشرفته ویندوز NT ادامه داد. در این مقطع مایکروسافت تصمیم گرفت Server SQL را شخصاً برای محصول جدید خود توسعه دهد. نام این محصول جدید SQL Server 4.2 بود که بعدها به ۴,۲۱ ارتقاء یافت.

پس از جدا شدن مایکروسافت و Sybase، شرکت اخیر موتور پایگاه داده خود را برای اجرا بر روی ویندوز NT توسعه داد (این محصول Enterprise Sybase Adaptive Server نامیده می شود) و مایکروسافت ویرایش ۶ و سپس ۶/۵ را عرضه کرد. ویرایش ۶/۵ این محصول که با قابلیت اجرا در محیط های NT و ویندوزهای ۹۵ و ۹۸ معرفی شد، بسیار وابسته به Sybase بود اما در ویرایش ۰.۷ تیم مایکروسافت بتدریج کد مزبور را دوباره نویسی کرد و آن را تغییر داد. این شرکت ضمن ساختاردهی مجدد هسته مرکزی موتور پایگاه داده، یک بهینه ساز پرس و جوی پیشرفته و پیچیده و یک موتور ذخیره سازی پایگاه داده پیشرفته را نیز معرفی کرد.

به صورت کلی، SQL Server یک نرم افزار سیستم مدیریت بانک های اطلاعاتی است که توسط شرکت مایکروسافت توسعه داده شده است. برخی از ویژگی های این سیستم مدیریت پایگاه داده ها به این شرح است:

۱. بانک اطلاعاتی رابطه ای
۲. امکان استفاده از Trigger, View, Stored procedure
۳. پشتیبانی از XML
۴. OLAP
۵. بسیار قدرتمند و بدون محدودیت حجم و تعداد رکورد

۲.۴ تکنولوژی LINQ

شرکت مایکروسافت، Language Integrated Query یا LINQ را همراه با NET Framework 3.5 معرفی کرد. LINQ، برنامه نویسان را قادر به query کردن منبع داده ها با استفاده از یک query مانند syntax با C# و VB.NET می کند. این منبع داده ها می توانند collection ها، بانک های اطلاعاتی SQL Server، XML و dataset ها باشند. به غیر از آنچه که از طرف Microsoft تامین می شود، LINQ گسترده هم هست. این بدین معناست که شما می توانید منابع داده ها را فراتر از آنچه که مایکروسافت ship میکند، query کنید. مثال هایی از چنین پیاده سازی هایی عبارتند از LINQ To Flickr، LINQ To Amazon، LINQ to Google، و غیره.

۲.۵ Ajax چیست

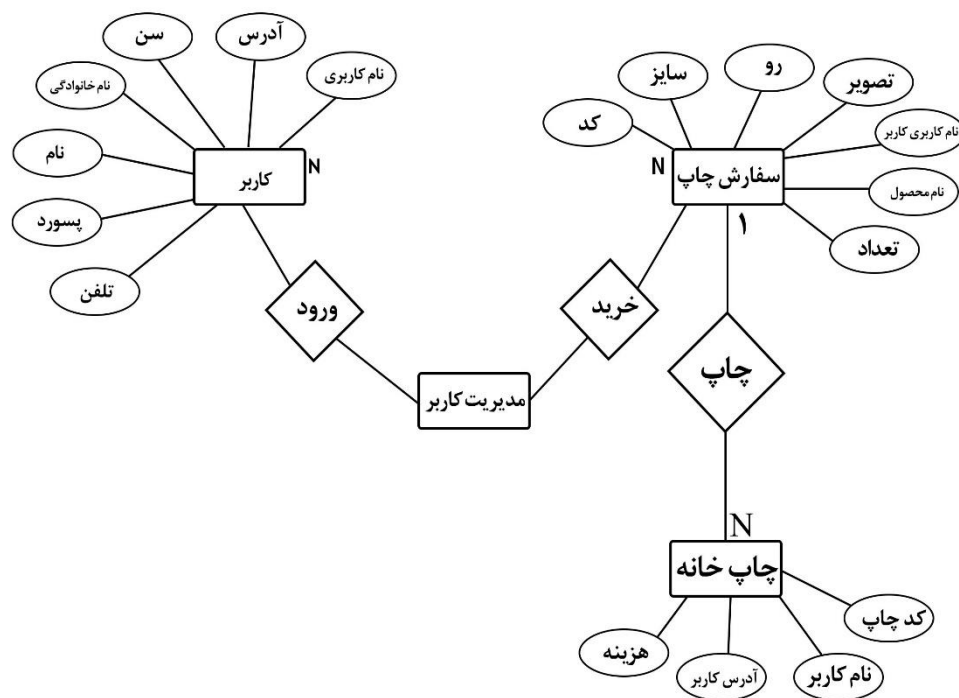
شاید نتوان Ajax را به سادگی توصیف کرد ولی میتوان در یک جمله گفت Ajax ما را قادر میسازد بخشهایی از صفحه را بدون بارگذاری کل صفحه (refresh) به روز رسانی کنیم. مسلماً این قابلیت است که خیلی ها به دنبال آن هستند زیرا بارگذاری مجدد صفحات وب همواره زمانبر و خسته کننده است ضمن اینکه باعث میشود پهنای باند کاربر نیز بیش از حد مصرف شود Ajax. تکنیکی برای ایجاد صفحات وب سریع و پویا میباشد Ajax. به صفحات وب این امکان را میدهد که به صورت غیر همزمان و تنها با تبادل اطلاعات اندکی با سرور، بخشی از صفحه را به روز رسانی کنند. به این ترتیب میتوان تنها بخش(هایی) از صفحه را بدون بارگذاری کل صفحه به روز رسانی کرد. در صورتی که صفحات معمولی باید کل صفحه را به منظور تغییر محتوا به روز رسانی کنند. بسیاری از سایتهای معروف از جمله Google Maps، Gmail، YouTube و Facebook به شکل وسیعی از این فن آوری استفاده می کنند. یکی از معروفترین مثالهای Ajax، سیستم Google Suggest است که به شکل بسیار زیبایی پیاده سازی شده است. شما عبارتی را در کادر جستجوی Google تایپ میکنید و Google بلافاصله با یک عملیات Ajax پیشنهادهایی را در مورد این عبارت به شما ارائه می کند.

فصل سوم

نمودار ها و جداول

۳.۱ نمودار ER

نمودار ER بیانگر ارتباط موجودیت (نهاد) های یک بانک اطلاعاتی است که کار مدل بندی داده های موجود را انجام میدهد. برای مثال محیط عملیاتی یک سامانه چاپ آنلاین را در نظر بگیرید. این محیط از نهادهای، کاربران، مدیر چاپخانه، مدیر سایت، پستی و ... تشکیل شده است. هر ارتباط بین دو نهاد کاربر و مدیر سایت، دارای مفهومی است که بین موجودیت های آن رابطه تعلق دارد و ارتباطات دیگر فاقد آن مفهوم هستند. در واقع این مفهوم دارای یک بار اطلاعاتی و داده ای است. رابطه ها میتوانند بین چند موجودیت نیز به اشتراک گذاشته شوند و مفهوم داده ای خود را بین چند موجودیت یا نهاد قرار دهند.

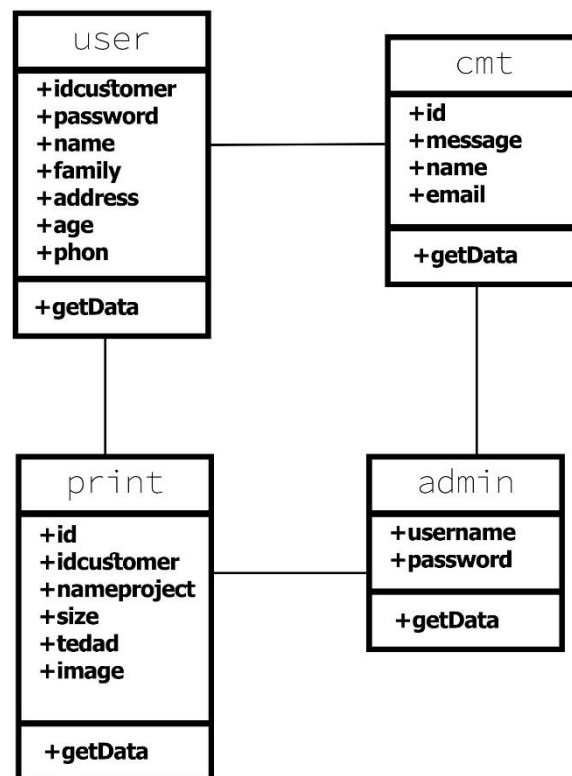


شکل 1.3 نمودار ER

۳.۲ نمودار کلاس

نمودار کلاس خصیصه ها (attribute) و عملیات (operation) یک کلاس و همچنین محدودیت ها (constraint) اعمال شده بر روی سیستم را توصیف می کند. نمودارهای کلاس به طور گسترده به منظور مدل سازی سیستم های شی گرا بکار گرفته می شوند، زیرا نمودار کلاس تنها دیاگرامی است که می تواند توسط زبان های شی گرا نگاشت (map) شود.

نمودار کلاس مجموعه ای از کلاس ها، interface ها، association ها، collaboration ها و محدودیت ها را به نمایش می گذارد. نمودار کلاس زیرمجموعه ی نمودار ساختاری (structural diagram) می باشد.



شکل 2.3 نمودار کلاس

۳.۳ پایگاه داده

بانک اطلاعاتی یا پایگاه داده به مجموعه ای از اطلاعات با ساختار منظم گفته می شود. این پایگاه های اطلاعاتی معمولاً در قالبی که برای دستگاه ها و رایانه ها قابل خواندن و قابل دسترسی باشند ذخیره می شوند.

پایگاه داده اس کیو ال چیست ؟ با گسترش طراحی وب سایت های پویا در شبکه جهانی وب نیاز به یک پایگاه داده و بانک اطلاعاتی بزرگ برای مدیریت محتوا احساس میشد. مدیریت پایگاه داده ها یک فرایند پیچیده است ، که به طور قابل توجهی با برنامه و زبان برنامه نویسی SQL این نیاز برطرف شده است. پایگاه داده در اصل مجموعه ای سازمان یافته از اطلاعات است. این واژه از دانش رایانه سرچشمه می گیرد ،اما کاربر وسیع و عمومی نیز دارد، این وسعت به اندازه ای است که مرکز اروپایی پایگاه داده (که تعاریف خردمندانه ای برای پایگاه داده ایجاد می کند) شامل تعاریف غیر الکترونیکی برای پایگاه داده می باشد. در این نوشتار به کاربرد های تکنیکی برای این اصطلاح محدود می شود.

یک تعریف ممکن این است که: پایگاه داده مجموعه ای از رکورد های ذخیره شده در رایانه با یک روش سیستماتیک (اصولی) مثل یک برنامه رایانه ای است که می تواند به کاربر پاسخ دهد. برای ذخیره و بازیابی بهتر، هر رکورد معمولاً به صورت مجموعه ای از اجزای داده ای یا رویداد ها سازماندهی می گردد.

cmt			
	Column Name	Condensed Type	Nullable
🔑	id	int	No
	message	nvarchar(MAX)	Yes
	name	nvarchar(50)	Yes
	email	nvarchar(50)	Yes

admin			
	Column Name	Condensed Type	Nullable
🔑	username	nvarchar(50)	No
	password	nvarchar(50)	No

user			
	Column Name	Condensed Type	Nullable
🔑	idcustomer	nvarchar(50)	No
	password	nvarchar(50)	Yes
	name	nvarchar(50)	Yes
	family	nvarchar(50)	Yes
	address	nvarchar(50)	Yes
	age	nvarchar(50)	Yes
	phon	nvarchar(50)	Yes

print			
	Column Name	Condensed Type	Nullable
🔑	id	int	No
	idcustomer	nvarchar(50)	Yes
	nameproject	nvarchar(50)	Yes
	size	nvarchar(50)	Yes
	tedad	nvarchar(50)	Yes
	image	nvarchar(MAX)	Yes

شکل 3.3 شرح جداول

فصل چهارم

پیاده سازی نرم افزار

۱,۴ سناریو ها

بعضی سناریو های مهندسی نرم افزار این پروژه:

اسم مورد کاربرد	ثبت نام
شماره	۱
هدف اصلی	ثبت نام کاربر (مدیر سایت)
نوع	اصلی
عامل	کاربر (مدیر سایت)
پیش شرط	قبلا ثبت نام نکرده باشد و تایید شده از طرف مجموعه باشد
روند اصلی	۱. کاربر وارد سامانه میشود ۲. به قسمت Register می رود ۳. فرم مشخص شده را با اطلاعات خود پر میکند و دکمه Register را می زند ۴. سیستم بررسی میکند که کاربر منحصر به فرد است یا نه ۵. سیستم کاربر را به صفحه شخصی (پنل کاربر) هدایت می کند
پس شرط	کاربر به سیستم اضافه میشود
روند جایگزین	اگر سیستم پس از بررسی متوجه شود که کاربر قبلا ثبت نام کرده پیغامی چاپ می کند.

جدول ۴-۱ سناریو ثبت نام

عضویت در سایت

نام :

نام خانوادگی :

ایمیل :

نام کاربری :

رمز عبور :

تکرار رمز عبور :

آدرس :

شکل ۴-۱ صفحه ثبت نام

توضیح: در این اتوماسیون دو نوع ثبت نام کاربر وجود دارد:

۱. ثبت نام کاربر(مدیر سایت) که پنلی جدا برای آن ساخته شده است.
۲. ثبت نام کاربر(بخش فنی و ادمین ها) که برای این قسمت هم پنلی جدا وجود دارد که در قسمت پنل ادمین کل این کاربر ها از طریق ادمین کل اضافه می شوند.

اسم مورد کاربرد	Add New News
شماره	۲
هدف اصلی	انتشار خبر جدید
نوع	اصلی
عامل	مدیران سایت
پیش شرط	مدیران به بخش پنل کاربران ورود کرده باشند و تصویر خود را بارگزاری کند و و متن خبر را منتشر کنند.
روند اصلی	<p>۱. کاربر وارد سامانه می شود</p> <p>۲. به قسمت Login می رود و وارد پنل می شود</p> <p>۳. روی گزینه انتشار خبر جدید کلیک می کند</p> <p>۴. مقادیر را پر کند تصویر را بارگزاری کند</p>
روند جایگزین	بعد از ارسال تصویر و متن بتواند آن را ویرایش کند

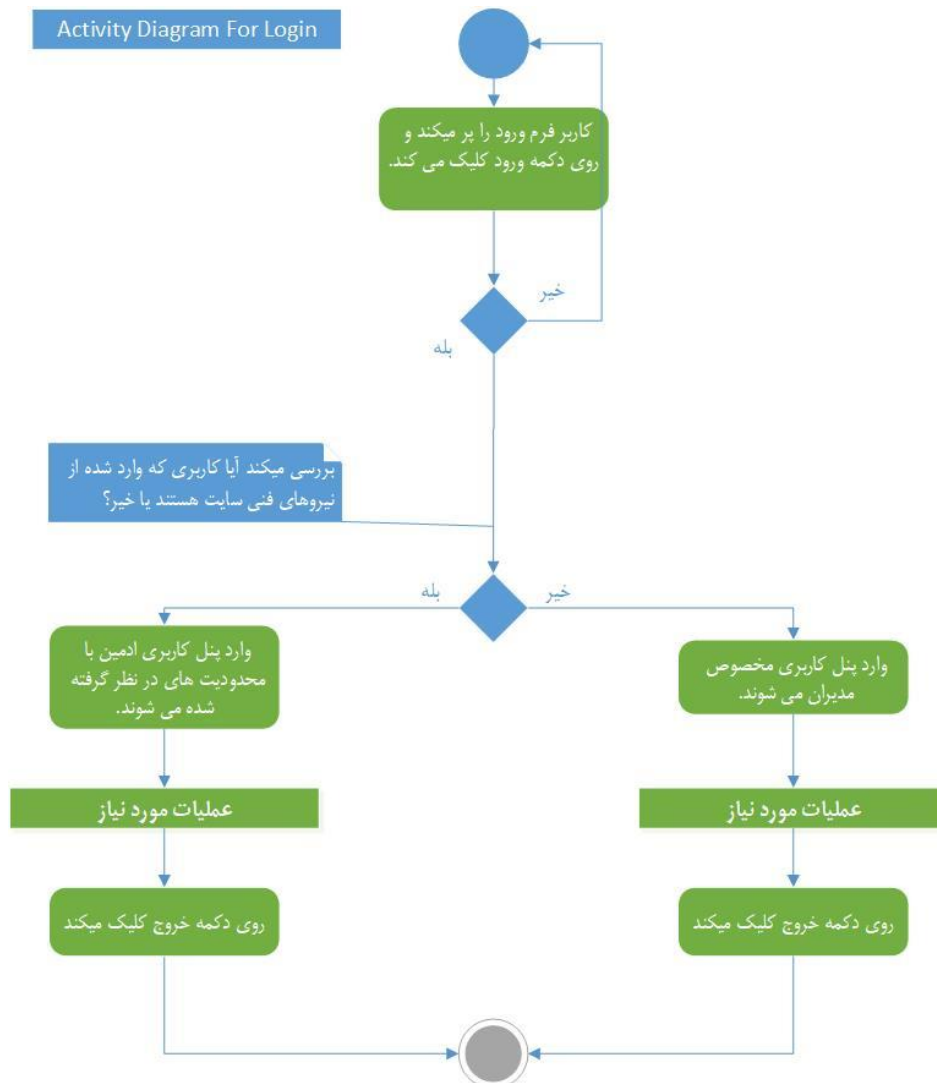
جدول ۴-۲ سناریو انتشار خبر جدید

اسم مورد کاربرد	نمایش خبر های قبلی
شماره	۳
هدف اصلی	دیدن خبر ها
نوع	اصلی
عامل	کاربر
پیش شرط	کاربر باید از قبل خبری را انتشار داده باشند
روند اصلی	<p>۱. کاربر وارد سامانه می شود</p> <p>۲. به قسمت Login می رود و وارد پنل می شود</p> <p>۳. در قسمت خبر های قبلی کلیک کند</p> <p>۴. فرمی بارگزاری می شود و تمام خبر های قبلی خود را میبیند.</p>
روند جایگزین	...

جدول ۳-۴ سناریو دیدن خبر های قبلی

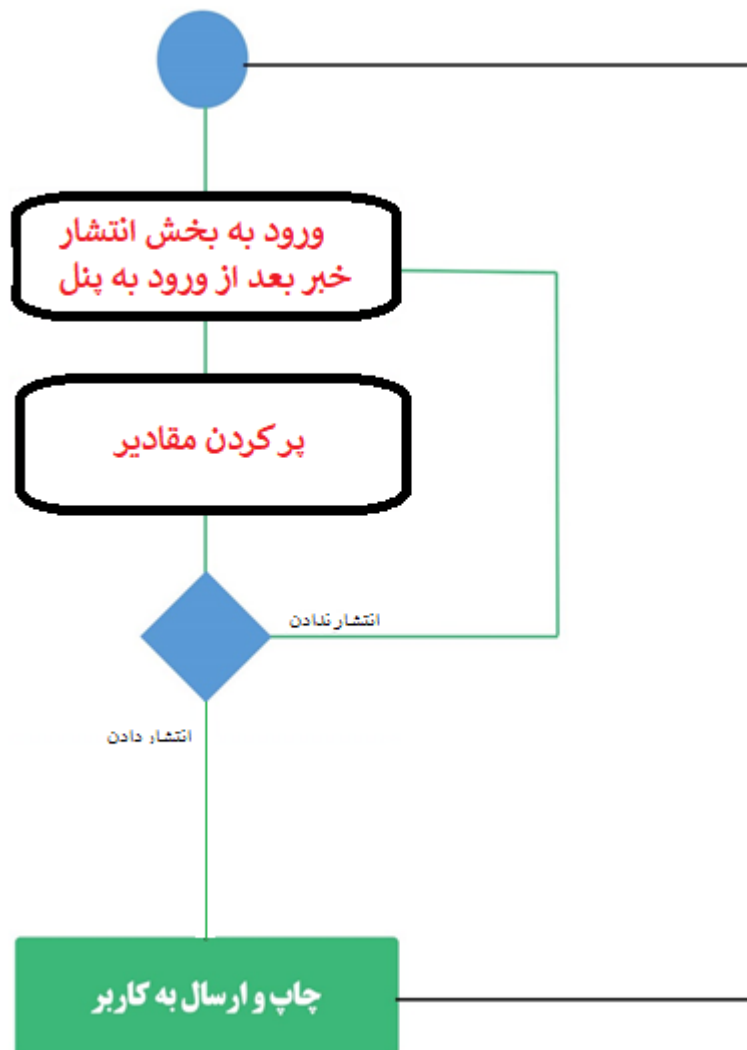
۴.۲ UML Activity Diagrama

در تصاویر زیر فلوچارت های مربوط به لاگین و ساخت پروژه و درخواست اصلاحات نمایش داده شده است که هر کدام روند طی شده برای رسیدن به عملیات را مرحله به مرحله شرح می دهد.



۴-۴ فلوچارت لاگین

در تصویر زیر فلوچارت نحوه ثبت خبر ترسیم شده است و مراحل ثبت خبر را مشاهده می کنید.



شکل ۴-۵ فلوچارت ثبت خبر

۴.۳ الگوی Repository

معنی لغوی رپوزیتوری مخزن و یا ذخیره کردن اطلاعات می باشد و شگردی جدید در برنامه نویسی محسوب می شود. الگوی Repository و Unit test (یونیت تست) در حقیقت لایه ای بین Data Access Layer و Bussiness Logic Layer می باشد، پیاده سازی این الگو شما را قادر خواهد کرد که برنامه های خود را در مقابل ذخیره داده ها و ساخت امکانات خودکار و یونیت تست ها ایمن و ایزوله و انعطاف پذیر نمایید و کد کمتری نوشته و بهره بیشتری از آن ببرید.

مزایا:

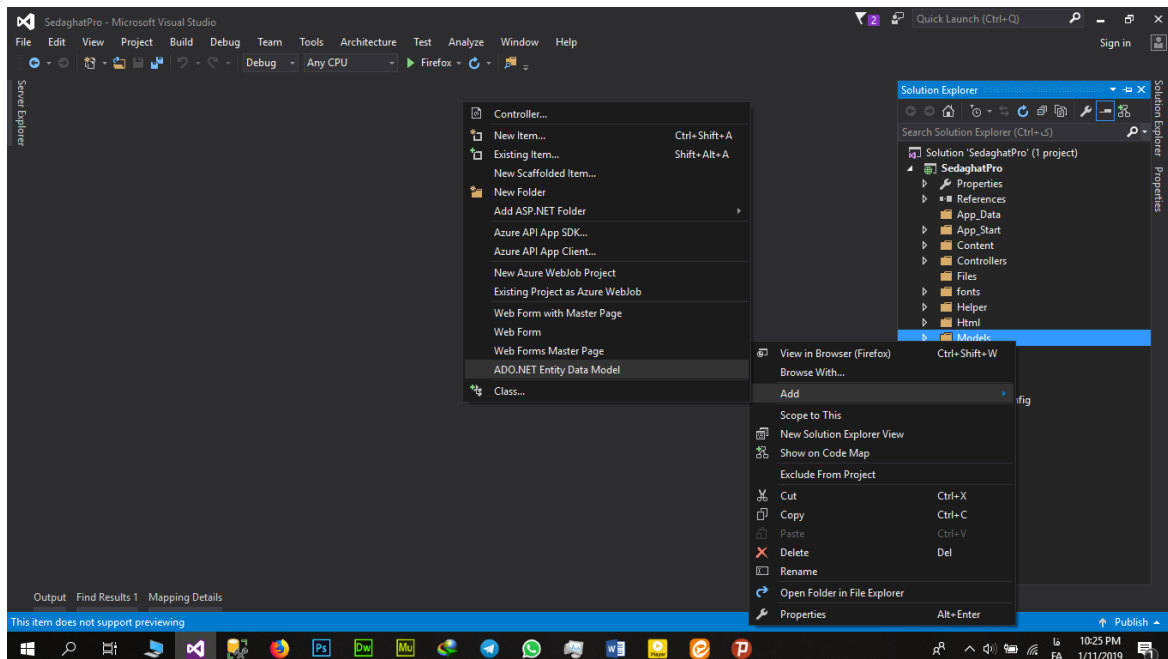
۱. بالا بردن خوانایی و نگهداری منطق تجاری با جداسازی دسترسی به دیتا
۲. کنترل مرکزی نسبت به دسترسی به دیتا
۳. امکان تعویض تکنولوژی مربوط به دسترسی به دیتا در آینده
۴. نوشتن تست های واحد (Unit Test) به جای تستهای یکپارچه (Integration Test)
۵. کاهش پیچیدگی در سطح کد و نیز کاهش کد نویسی

۴.۴ Entity Framework چیست

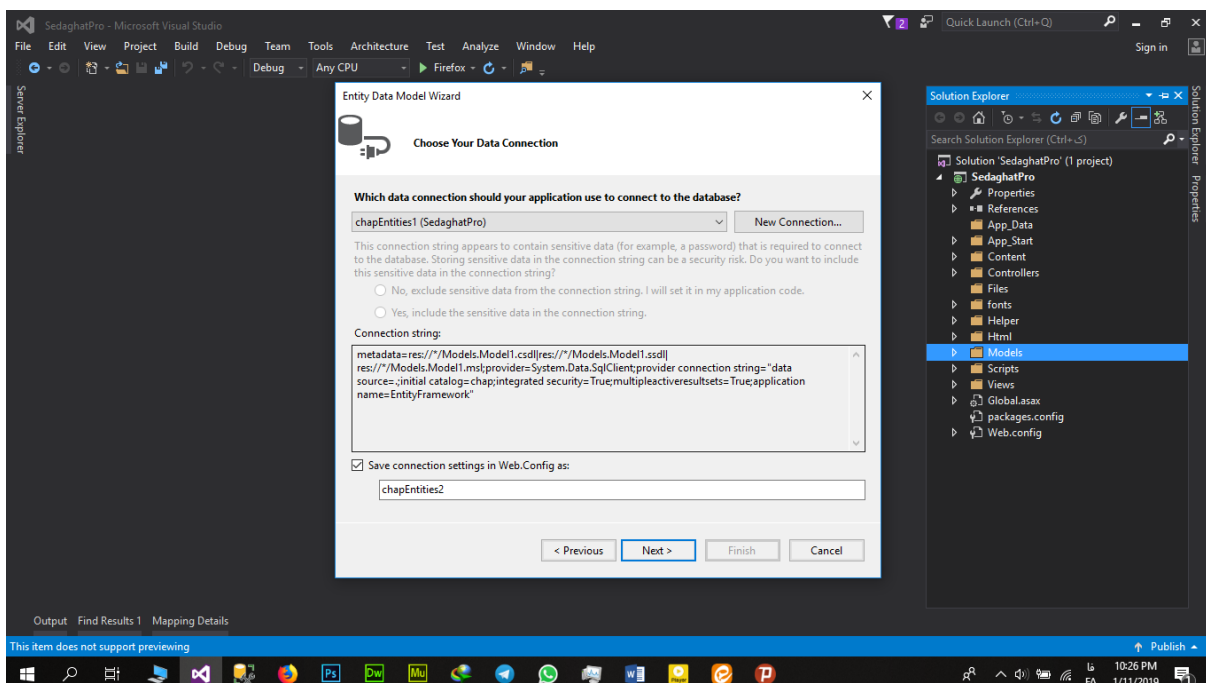
Entity Framework به سادگی به داده ها دسترسی دارد و به طور مستقیم با پایگاه داده کار می کند. در این روش احتیاجی به نوشتن کد اضافی برای مدیریت داده ها و تعامل آنها با پایگاه داده نیست. همچنین نیازی به ایجاد پایگاه داده و جدول داده ها به طور دستی وجود ندارد زیرا Entity Framework آن را به طور اتوماتیک برای شما ایجاد می کند.

Entity Framework بهترین انتخاب برای برنامه های Asp.net MVC می باشد. بسیاری از توسعه دهندگان وب که در زمینه asp.net MVC کار می کنند همیشه Entity Framework را برای دسترسی به پایگاه داده ترجیح می دهند.

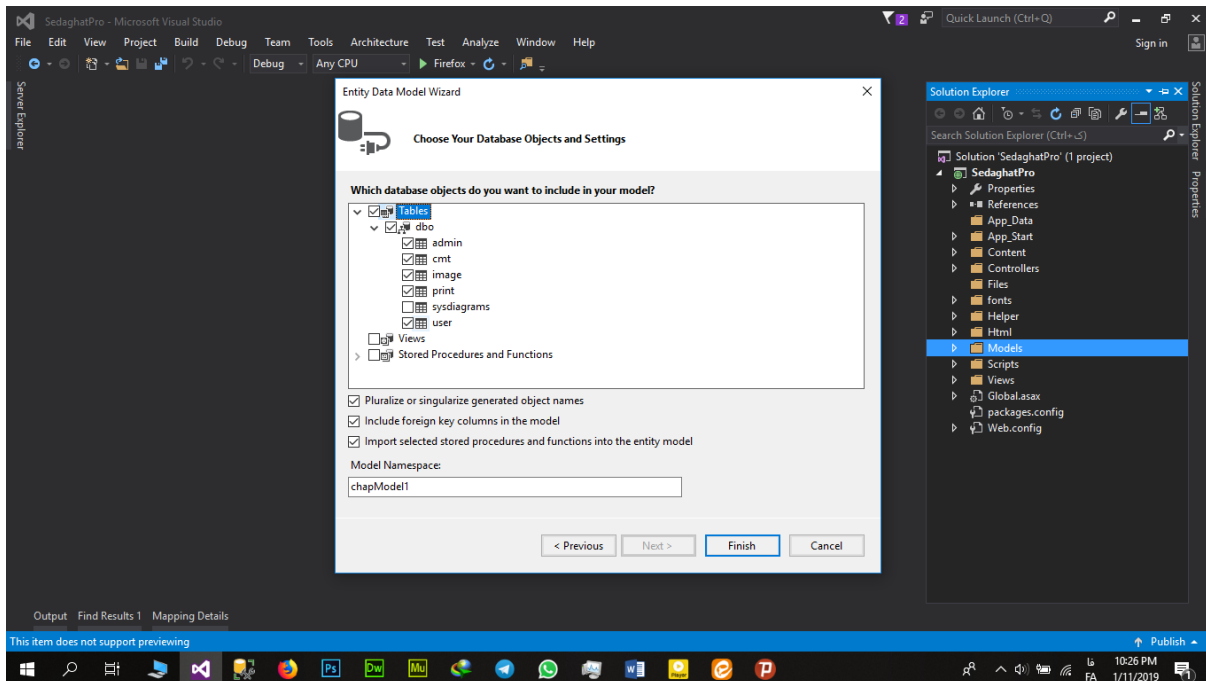
۴.۴.۱ نحوه ایجاد Entity Framework



۶،۴ ایجاد Entity Framework مرحله ۱

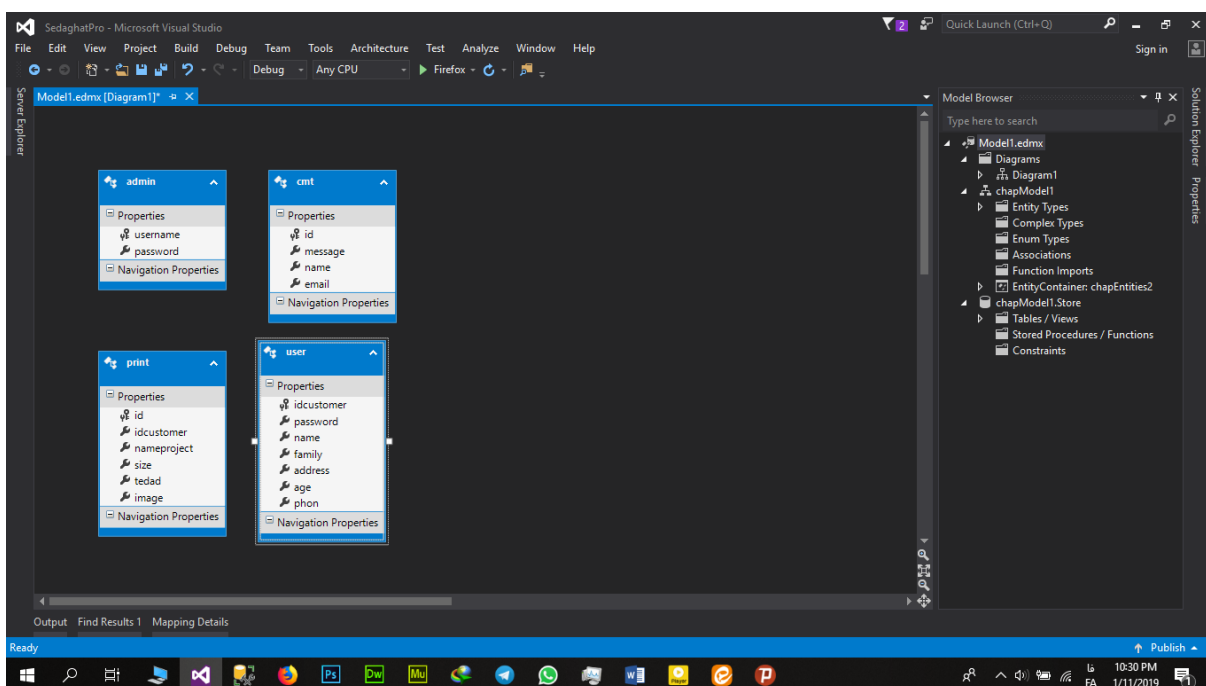


شکل ۷،۴ ایجاد Entity Framework مرحله ۲



شکل ۸,۴ ایجاد Entity Framework مرحله ۳

جدول هایمان را انتخاب می کنیم.



شکل ۹,۴ ایجاد Entity Framework مرحله ۴

توضیح: ایجاد EntityFramWork

۱. روی پوشه Model کلیک راست میکنیم و گزینه Add ، سپس گزینه Ado.Net Entity Data Model کلیک میکنیم.(شکل ۴-۶)
۲. پایگاه داده خود را انتخاب میکنیم نام EntityFramWork را انتخاب میکنیم(شکل ۴-۷)
۳. از پایگاه داده جداول مورد نظر خودمان را به پروژه اضافه میکنیم(شکل ۴-۸)
۴. تمام جداولی که انتخاب کرده ایم وارد پروژه ایمان شده اند.(شکل ۴-۹)

۴.۴.۲ مزیت های Entity Framework

۴.۴.۲.۱ آسان کردن تغییر بانک اطلاعاتی

وقتی شما با یک بانک اطلاعاتی مثل SQL Server کار می کنید و بعد از گذشت یک مدت میخواهید بانک اطلاعاتی تون را از SQL Server به مثلاً SQLLight تغییر بدید Entity Framework Core این کار را برای ما به راحتی انجام می دهد.

۴.۴.۲.۲ کوئری نویسی با استفاده از LINQ

با بهره گیری از Entity Framework می توانیم با استفاده از LINQ کوئری نویسی انجام دهیم. و با کوئری نویسی از طریق LINQ خطر حمله SQL Injection را تقریباً به صفر برسانیم..

۴.۴.۲.۳ چند سکویی بودن

یکی دیگر از مزیت های Entity Framework Core این است که چند سکویی هست و قابلیت اجرا روی Mac و Linux را داراست.

۴.۴.۲.۴ پشتیبانی از اکثر بانک های اطلاعاتی

Entity Framework Core نسبت Entity Framework 6 از بانک های اطلاعاتی بیشتری پشتیبانی می کند و در نسخه Entity Framework Core پشتیبانی از بانک های اطلاعاتی NoSql هم افزوده شده است.

۴.۵ اعتبار سنجی اطلاعات ورودی در فرم‌های ASP.NET MVC

زمانیکه شروع به دریافت اطلاعات از کاربران کردیم، نیاز خواهد بود تا اعتبار اطلاعات ورودی را نیز ارزیابی کنیم. در ASP.NET MVC، به کمک یک سری متادیتا، نحوه‌ی اعتبار سنجی، تعریف شده و سپس فریم ورک بر اساس این ویژگی‌ها، به صورت خودکار اعتبار اطلاعات انتساب داده شده به خواص یک مدل را در سمت کلاینت و همچنین در سمت سرور بررسی می‌نماید. این ویژگی‌ها در اسمبلی DataAnnotations.dll قرار دارند که به صورت پیش فرض در هر پروژه جدید ASP.NET MVC لحاظ می‌شود.

به سادگی از طریق مدل خواندن قوانین مربوط به داده‌ها را برای این برنامه نشان می‌دهد، و باعث می‌شود که کد را راحت تر نگه دارید. در زیر چندین ویژگی معتبر ساخته شده در معتبر وجود دارد:

[CreditCard]: تأیید می‌کند که اموال دارای یک کارت اعتباری است.

[Compare]: دو خواص را در یک بازی مدل معتبر می‌کند.

[EmailAddress]: اعتبار اموال یک فرمت ایمیل دارد.

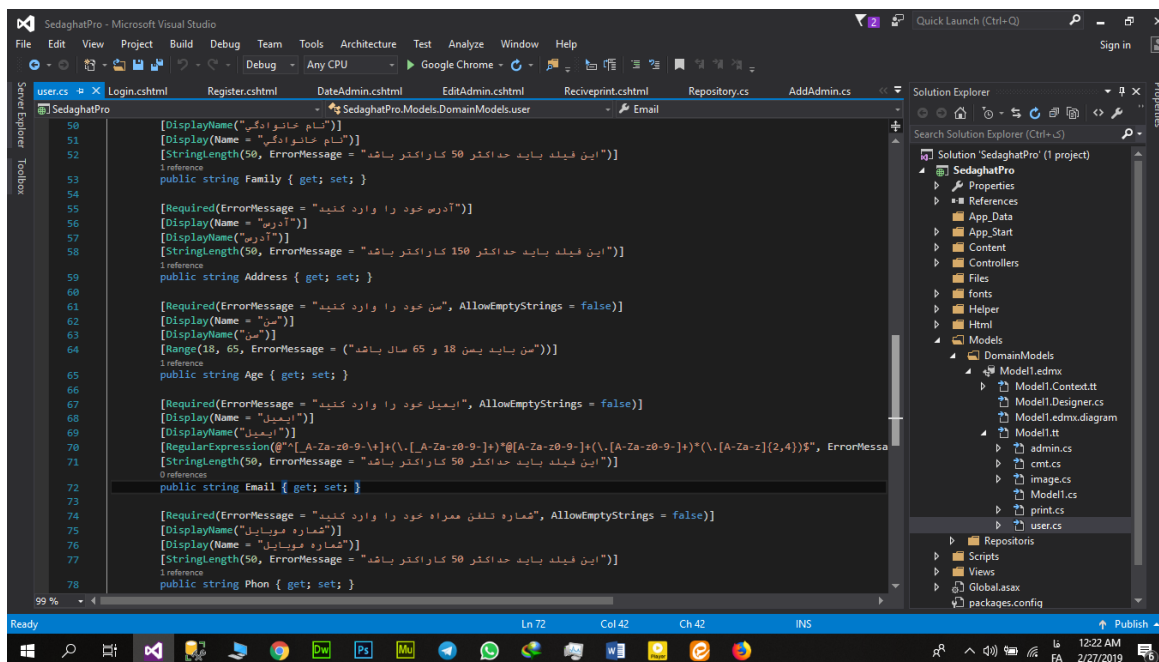
[Range]: ارزش املاک را در محدوده داده شده تأیید می‌کند.

[RegularExpression]: تأیید می‌کند که داده‌ها با عبارات منظم مشخص مطابقت دارند.

[Required]: حتماً باید یک مقدار وارد شود.

[StringLength]: تأیید می‌کند که یک دارایی رشته دارای حداکثر طول داده شده است.

[Url]: تأیید می‌کند که اموال یک فرمت URL دارد.



شکل ۱۰,۴ اعتبارسنجی ثبت نام کاربران

توضیح: نحوه ایجاد و نوشتن اعتبار سنجی سمت کاربر

۱. روی جدول مورد نظرمان که با استفاده از EntityFramWork اضافه کرده ایم، کلیک کنید.
۲. `using System.ComponentModel.DataAnnotations;` را به کتابخانه هایمان اضافه میکنیم.
۳. برای هر فیلد از که میخواهیم اعتبار سنجی مورد نظرمان را مینویسیم.

۴.۶ Razor چیست

Razor یا رازور (ریزور) یک انجین تبدیلی کدهای سرور ساید یا سمت سرویس دهنده Asp.net به چ تی ام است که از نسخه 3 MVC به بعد از طرف ماکروسافت ارائه شده و نسبت به صفحات قدیم asp.net که عموماً با پسوند aspx شناخته می شوند از chtml استفاده می نمایند. در این مدل صفحات که برای برنامه نویسان Asp.net MVC توصیه شده اما برنامه نویسان Asp.net Web Form هم می توانند از آن استفاده کنند، کار با Razor بسیار راحت تر شده و شما می توانید کد کمتر نوشته و بهره بیشتری از مطالب خود ببرید و در عین سادگی بسیار حرفه ای تر کد نویسی کنید ، سرعت بارگذاری یا رندر شدن

این صفحات از صفحات قدیمی aspx بالاتر است و دیگر از کنترل های <asp: و view state خبری نیست و همین امر در مصرف حافظه و ... تاثیر گذار است و Performance سایت را بالاتر می برد.

به نمونه کد زیر دقت کنید:

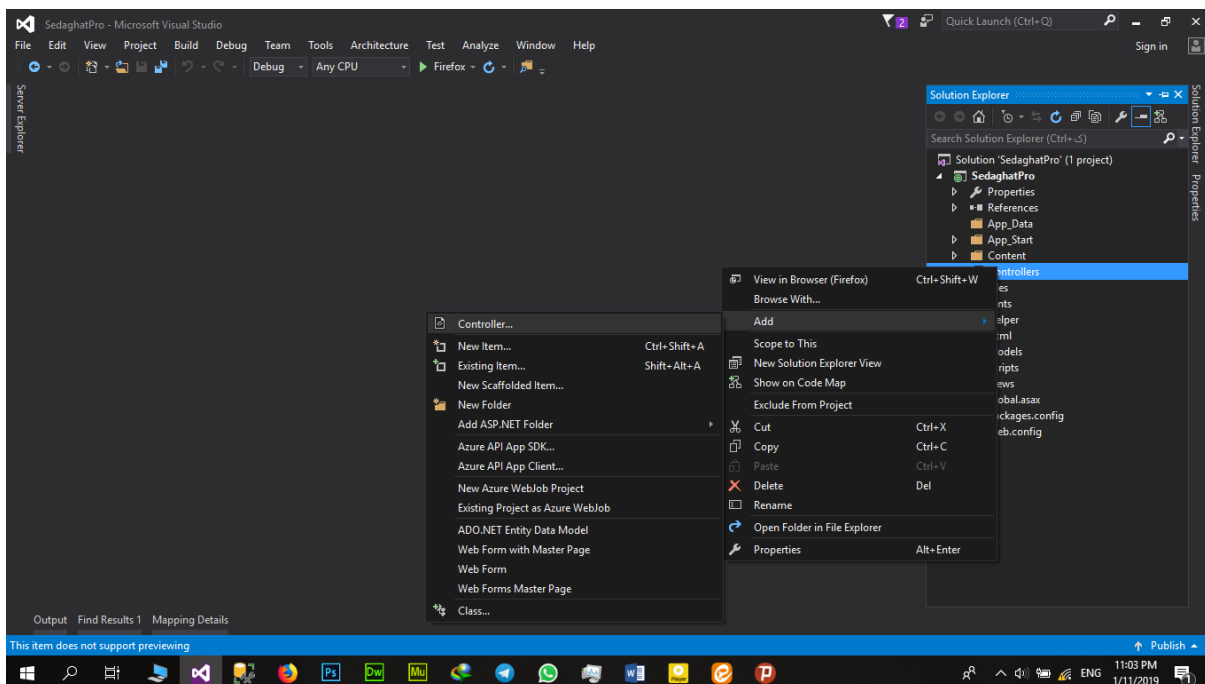
ساخت یک Text Box توسط صفحات قدیمی:

```
<Textbox server="runt" name=""/>
```

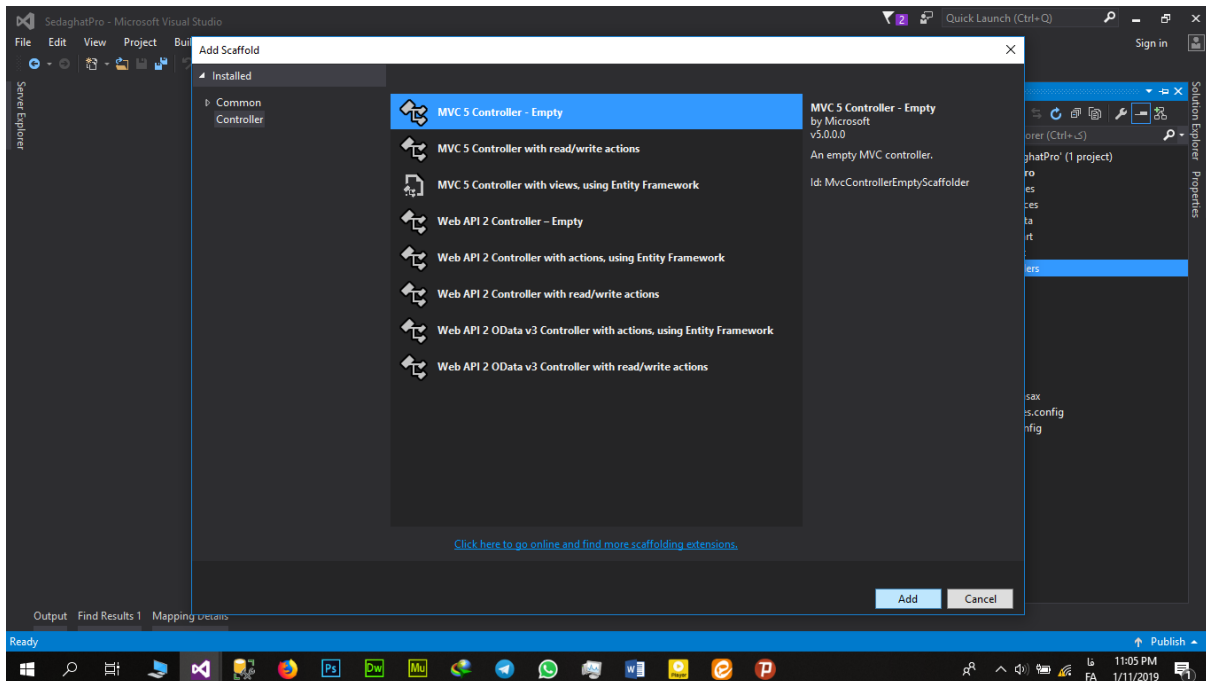
ساخت یک Text Box توسط صفحات Razor:

@Html.TextBox(name)

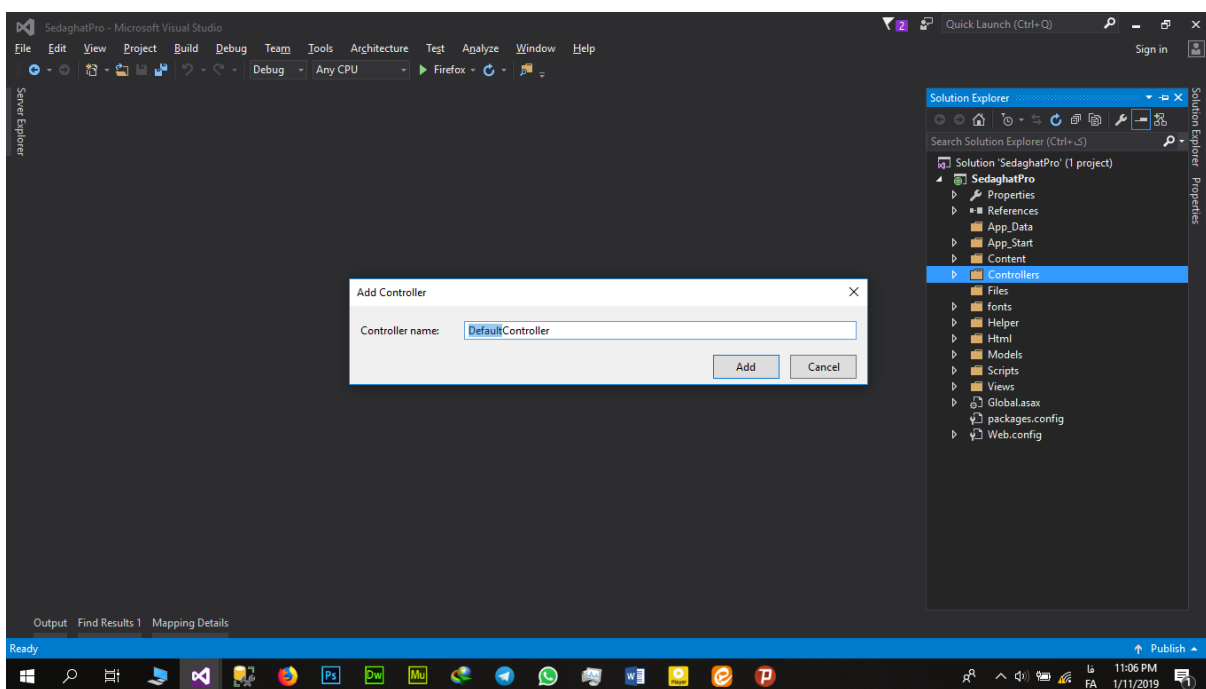
۴.۷ نحوه ایجاد controller



شکل ۱۱،۴ نحوه ایجاد contoroller مرحله ۱



شکل ۱۲،۴ نحوه ایجاد controller مرحله ۲



شکل ۱۳،۴ نحوه ایجاد controller مرحله ۳

توضیح: ایجاد Controller

۱. روی پوشه Controller کلیک راست میکنیم و گزینه Add ، سپس روی Controller کلیک میکنم (شکل ۴-۱۱)
۲. از پنجره باز شده MVC5 Controller-Empty را انتخاب میکنیم. (شکل ۴-۱۲)
۳. اسم کنترلر را انتخاب میکنیم، حتما باید متن Controller آخر اسممان باشد. (شکل ۴-۱۳)
۴. کنترلر ایجاد شد.

۴.۸ چیست ActionResult

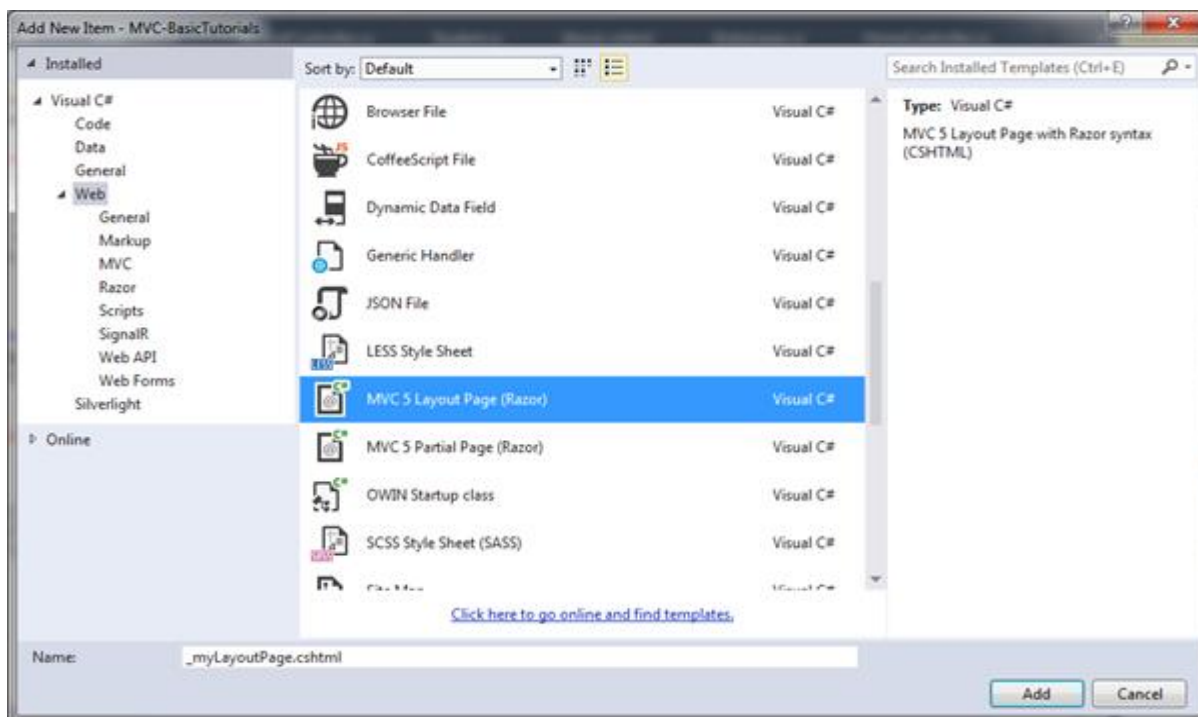
ActionResult یک کلاس انتزاعی برای نمایش خروجی به سمت کاربر در فرمت های مختلف که به عنوان نتیجه View توسط کنترلر (controller) را برمی گرداند ActionResult. در داخل کنترلر تعریف می شود و کنترلر را به سمت کاربر (مروگر) برمی گرداند.

در زیر انواع ActionResult های مهم و کاربردی ASP.Net MVC وجود دارد که براساس نیاز کاربر در خروجی استفاده می شود:

1. ViewResult
2. PartialViewResult
3. EmptyResult
4. RedirectResult
5. RedirectToRouteResult
6. JsonResult
7. JavaScriptResult
8. ContentResult
9. FileContentResult
10. FileStreamResult

۴.۹ ایجاد Layout page

برای ایجاد نمای جدید طرح بندی در ویژوال استودیو، روی پرونده مشترک کلیک راست کنید -> انتخاب افزودن بر روی مورد جدید کلیک کنید. در کادر محاوره ای Add New Item، صفحه MVC 5 Layout Page (Razor) را انتخاب کنید و عنوان صفحه طرح را به عنوان «MyLayoutPage.cshtml» قرار دهید و روی افزودن کلیک کنید.



شکل ۱۴،۴ نحوه ایجاد Layout page مرحله ۱

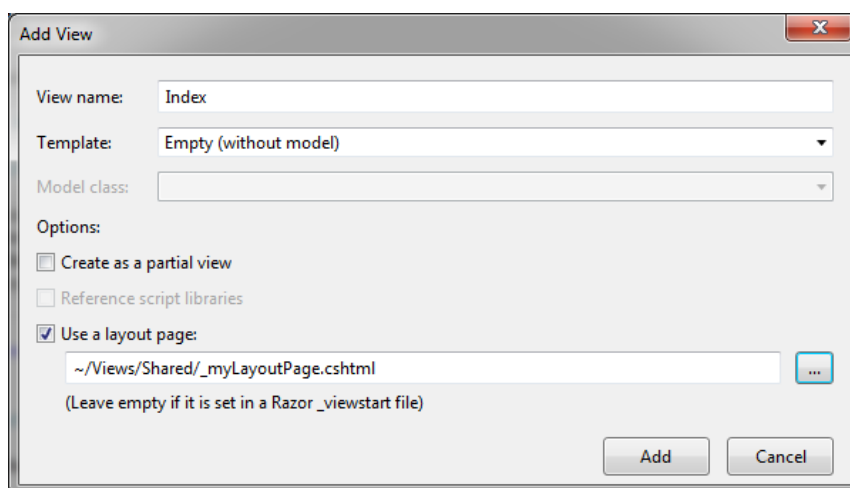
شما `_myLayoutPage.cshtml` را همانطور که در زیر نشان داده شده مشاهده خواهید کرد.

```
_myLayoutPage.cshtml:
<!DOCTYPE html>
<Html>
<head>
  <meta name="viewport" content="width=device-width" />
  <title>@ViewBag.Title</title>
</head>
<body>
  <Div>
    @RenderBody()
  </div></body>
</html>
```

اکنون، تگ <footer> را با RenderSection ("footer"true) همراه با برخی از یک ظاهر طراحی شده که در زیر نشان داده شده است اضافه کنید. لطفا توجه داشته باشید که ما این بخش را در صورت لزوم ساخته ایم. این به معنای هر دیدگاهی است که از MyLayoutPage_ به عنوان نمایه طرح آن استفاده می کند باید شامل بخش پایینی باشد.

```
_myLayoutPage.cshtml:
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width" />
<title>@ViewBag.Title</title>
</head>
<Body>
  <div>
    @RenderBody ()
  </div>
</body>
</html>
```

حالا، اجازه دهید از این _myLayoutPage.cshtml با نمای Index از HomeController استفاده کنیم. شما می توانید یک نمایه خالی Index را با کلیک راست بر روی روش Action Index HomeController اضافه کنید و Add View را انتخاب کنید. خالی را به عنوان یک الگوی داربست و _myLayoutPage.cshtml را به صورت طرح انتخاب کنید و روی افزودن کلیک کنید.



شکل ۱۵,۴ نحوه ایجاد Layout page مرحله ۲

این کار Index.cshtml را به صورت زیر نشان می دهد.

Index view:

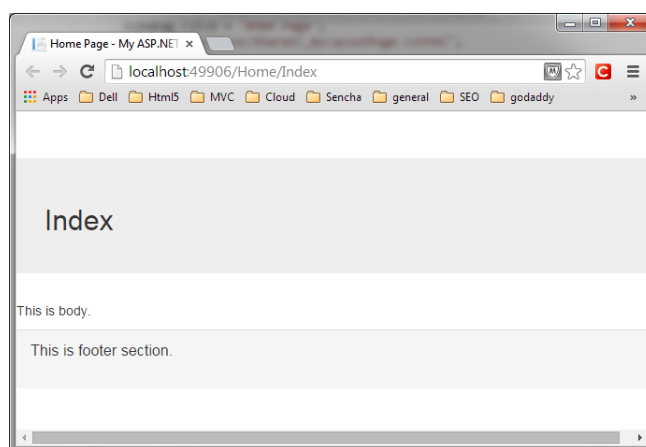
```
@{
    ViewBag.Title = "Home Page";
    Layout = "~/Views/Shared/_myLayoutPage.cshtml";
}
<h2>Index</h2>
```

بنابراین اکنون، ما نمایه Index ایجاد کرده ایم که از _myLayoutPage.cshtml ما به عنوان یک نمایه طرح استفاده می کند. حالا ما بخش افزونه را با برخی از یک ظاهر طراحی خواهیم کرد زیرا _MyLayoutPage نیاز به بخش پایین صفحه دارد.

Index view:

```
@{
    ViewBag.Title = "Home Page";
    Layout = "~/Views/Shared/_myLayoutPage.cshtml";
}
<div class="jumbotron">
    <h2>Index</h2>
</div>
<div class="row">
    <div class="col-md-4">
        <p>This is body.</p>
    </div>
    @section footer {
        <p class="lead">
            This is footer section.
        </p>
    }
</div>
```


در حال حاضر، برنامه را اجرا کنید و نمایه نمایه را مشاهده خواهید کرد، شامل قسمت بدن و پائین است که در زیر نشان داده شده است.



شکل ۱۶,۴ نحوه ایجاد Layout page مرحله ۳

بنابراین، شما می توانید نمایش طرح بندی جدید با روش های مختلف Render ایجاد کنید.

۴.۱۰ متد های ActionVerbs

در این بخش، شما در مورد ویژگی انتخابگر ActionVerbs یاد خواهید گرفت.

انتخاب ActivityVerbs هنگامی که شما می خواهید برای کنترل انتخاب روش عمل بر اساس یک روش درخواست Http استفاده کنید استفاده می شود. به عنوان مثال، شما می توانید دو روش مختلف عمل با همان نام را مشخص کنید، اما یک روش عمل به HTTP درخواست دریافت پاسخ می دهد و یک روش عمل دیگر به درخواست ارسال HTTP پاسخ می دهد.

چارچوب MVC پشتیبانی از ActionVerbs مختلف، مانند HttpGet، HttpPost، HttpPut، HttpDelete، HttpPatch و HttpOptions است. شما می توانید این ویژگی ها را به روش عمل برای نشان دادن نوع درخواست Http از روش عمل پشتیبانی کنید. اگر شما هیچ ویژگی را اعمال نکنید، آن را پیش فرض به عنوان درخواست GET می دانید.

Http method	Usage
GET	To retrieve the information from the server. Parameters will be appended in the query string.
POST	To create a new resource.
PUT	To update an existing resource.
HEAD	Identical to GET except that server do not return message body.
OPTIONS	OPTIONS method represents a request for information about the communication options supported by web server.
DELETE	To delete an existing resource.
PATCH	To full or partial update the resource.

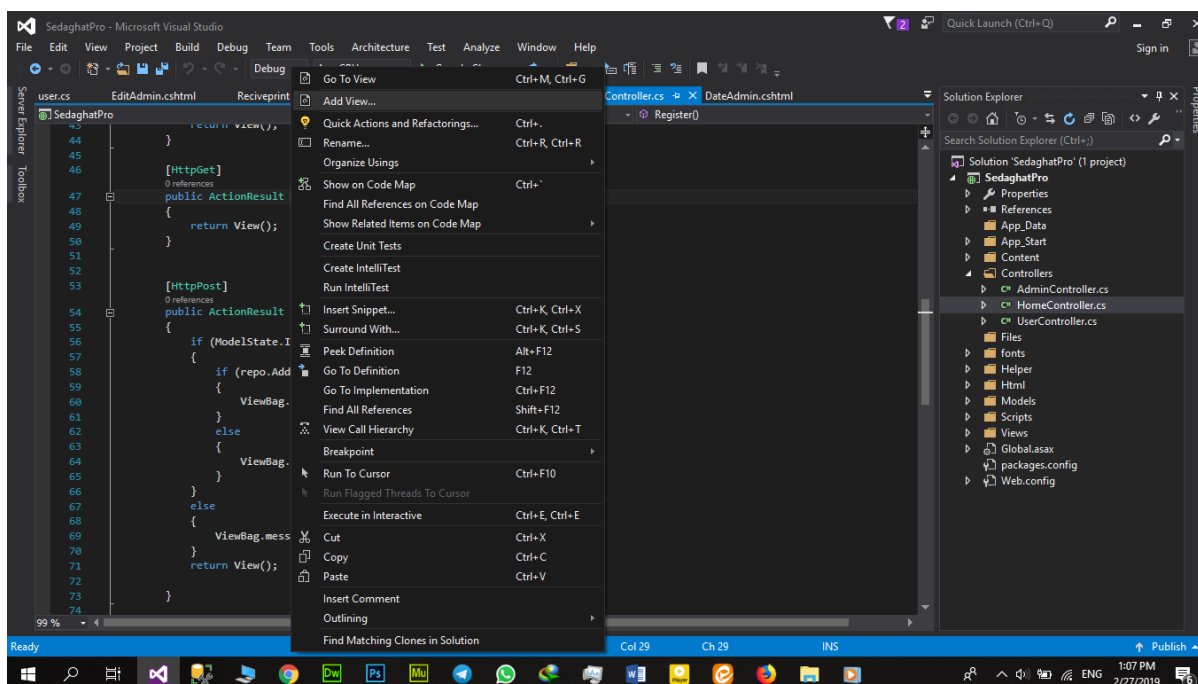
شکل ۱۷,۴ انواع ActionVerbs

۴.۱۱ متد [Http post] [Http get]

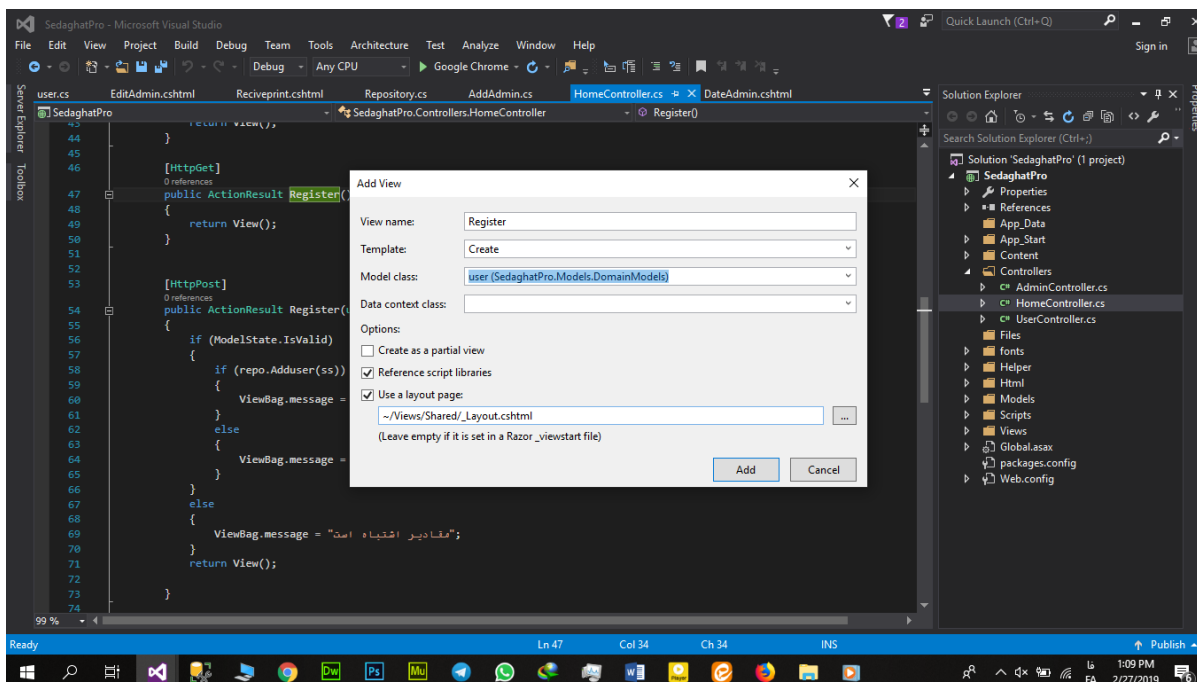
HttpGet و HttpPost هر دو روش ارسال داده های مشتری یا فرم داده ها به سرور هستند HTTP. یک پروتکل انتقال HyperText است که برای ارسال و دریافت اطلاعات بین سرویس گیرنده و سرور با استفاده از صفحات وب طراحی شده است. ویژگی های HTTPGET و HTTPPOST پارامترهای درخواست را به عنوان جفت کلید و ارزش در درخواست HTTP رمزگذاری می کنند. پروتکل HttpGet و پروتکل HttpPost سازگاری عقب مانده دارند. مثال زیر را پیدا کنید شما ممکن است دو نسخه روش جستجوی را بخواهید، یکی که فرم جستجو را ارائه می دهد و دیگری که درخواست را هنگامی که این فرم ارسال شده است را پردازش می کند. هر زمان درخواست POST / Value / Search دریافت می شود، دعوت کننده عمل لیستی از تمامی روش های کنترل کننده را مطابق با نام Action Action ایجاد می کند. در این مورد، شما می توانید لیستی از دو روش را بدست آورید. بلافاصله، فراخوانی به تمام نمونه های ActionSelectorAttribute مربوط به هر روش می پردازد و روش IsValidForRequest را در هر یک از آنها جستجو می کند. اگر هر ویژگی به درستی برسد، روش برای عمل فعلی معتبر است. بگذارید یک مورد را در نظر بگیریم هنگامی که از متد اول درخواست می کنید اگر بتواند درخواست POST را اداره

کند، با اشتباه پاسخ می دهد زیرا فقط درخواست های GET را انجام می دهد. روش دوم به درستی پاسخ می دهد زیرا می تواند درخواست POST را اداره کند و آن را برای رسیدگی به درخواست درخواست پست انتخاب کرده است. در حالی که انجام این عمل، هیچ روشی یافت نمی شود که مطابق با این معیارها باشد، فراخوانی روش HandleUnknownAction را بر روی Controller فراخوانی می کند، نامی از عمل گم شده را فراهم می کند. اگر بیش از یک روش عمل برآورده شده این معیارها پیدا شود، InvalidOperationException پرتاب می شود.

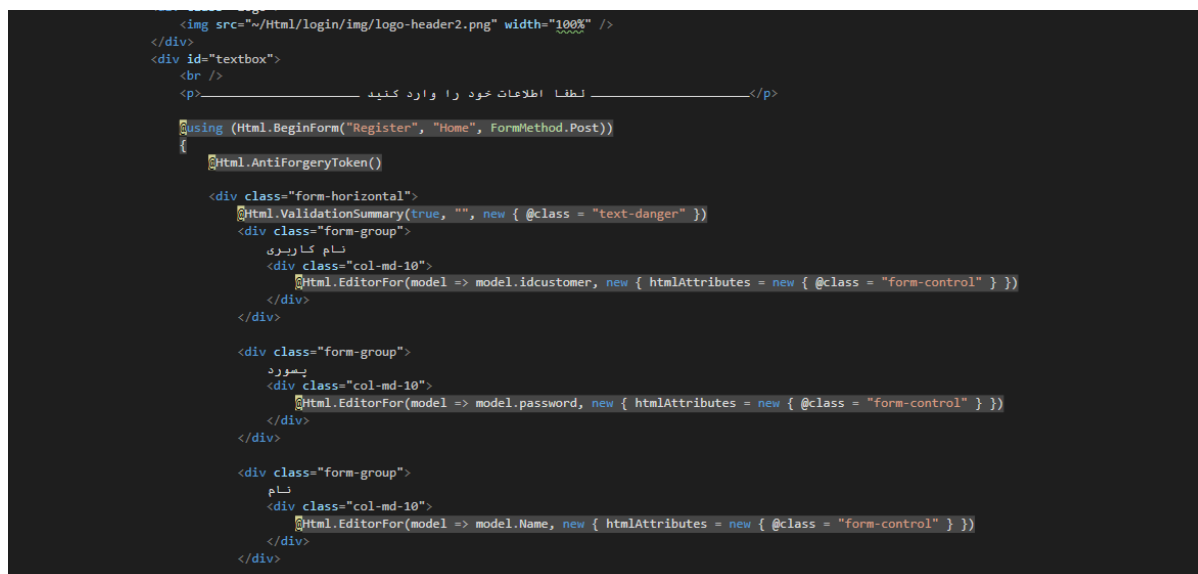
۴.۱۲ درست کردن view برای ثبت نام



شکل ۱۸،۴ ایجاد View مرحله ۱



شکل ۱۹،۴ ایجاد View مرحله ۱



شکل ۲۰،۴ ایجاد View مرحله ۲

توضیح: ایجاد view

۱. روی ActionResult کلیک راست میکنیم و گزینه Add View را انتخاب میکنیم.
(شکل ۴-۱۸)

۲. سپس کادری باز می شود، اسم View، Template را روی Create می گذاریم و ModelClass هم روی جدولی که قرار ثبت نام صورت بگیرد، سپس گزینه یی را میزنیم.(شکل ۴-۱۹)

۳. view ایی با فرمت CSHTML برایمان میسازد.

۴. طراحی هم برعهده خودمان است، می توان با استفاده از Latyot صفحه view را ساده تر کنیم.

```
public bool Adduser(SedaghatPro.Models.DomainModels.user entity, bool autoSave = true)
{
    try
    {
        db.users.Add(entity);

        if (autoSave)
            return Convert.ToBoolean(db.SaveChanges());
        else
            return false;
    }
    catch
    {
        return false;
    }
}
```

کد ثبت نام کاربران Repository

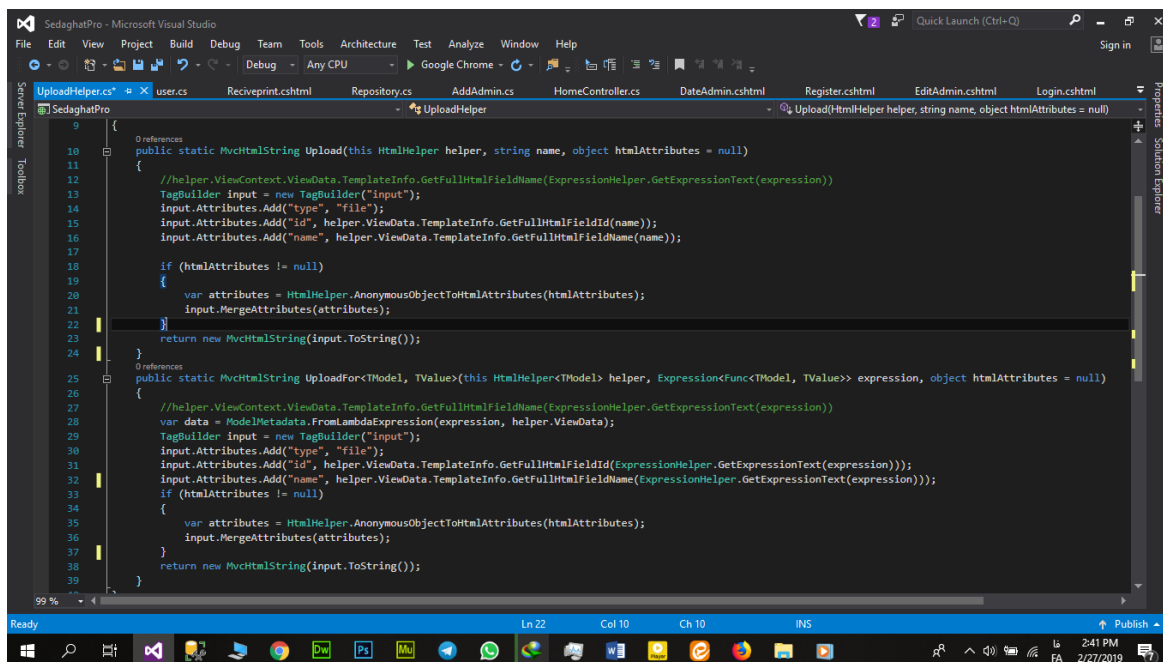
```
[HttpGet]
public ActionResult Register()
{
    return View();
}
[HttpPost]
public ActionResult Register(user ss)
{
    if (ModelState.IsValid)
    {
        if (repo.Adduser(ss))
        {
            ViewBag.message = "شد ثبت";
        }
        else
        {
            ViewBag.message = "نشد ثبت";
        }
    }
    else
    {
        ViewBag.message = "است اشتباه مقادیر";
    }
    return View();
}
```

کد ثبت نام Controller

۴.۱۳ متد htmlHelper

htmlHelper ها برای ایجاد ساختار و ایجاد کنترل های HTML مورد استفاده قرار می گیرند. HTML Helper متدی است، که یک رشته HTML را برمی گرداند. سه نوع از HTML Helper ها وجود دارد:

۱. **Inline HTML Helpers**: توسط تگ `@helper` از سینتکس Razor ایجاد میشود. این helper می تواند تنها در همان View مورد استفاده قرار گیرد.
۲. **Built-In HTML Helpers**: این HTML helper ها، Extension Method در کلاس `HtmlHelper` هستند. و به سه قسمت طبقه بندی میشوند:
 - **Standard HTML Helpers**: برای ایجاد رایج ترین تایپها از المنتهای HTML استفاده می شوند
 - **Strongly Typed HTML Helpers**: در کدهای HTML از طریق `Model Class Propertie` ها تولید میشود. این helper در `lambda expression` کار میکند.
 - **Templated HTML Helpers**: این helper ها ملزم به گفتن اینکه، چطور از المنت های html برای اجرا پروپرتی ها در کلاس های مدل استفاده کنیم، هستند.
۳. **Custom HTML Helpers**: شما همچنین می توانید متد های `Helper` سفارشی خود را با ایجاد یک `Extension Method` در کلاس `HtmlHelper` و یا با ایجاد متدهای استاتیک در داخل یک کلاس `utility` ایجاد کنید.



شکل ۲۱،۴ ایجاد یک Html Helper با استفاده از Extension Method

نکته :

۱- شی ایی از repository باید داخل پروژه نوشته شود

`using SedaghatPro.Models.Repository;`

`Repository repo = new Repository();`

۲- استفاده از مدل های دیگر هم باید namespace آن نوشته شود

`using SedaghatPro.Models.DomainModels;`

لینک در Reyzor syntax به روش زیر نوشته می شود

`@Html.ActionLink("نام ثبت", "Register", "Home", null)`

۳- قسمت اول متن لینک ، قسمت دوم اسم Action، قسمت سوم اسم Contoroller نوشته می شود.

۴- در تمامی صفحاتی که به آن ورود میکنند بالای صفحه این کد نوشته شده است:

```
if (Session["idcustomer"] == null)
{
    Response.Redirect("~/Home/Loginuser");
}
```

```
}  
if (Session["username"] == null)  
{  
    Response.Redirect("~/Home/LoginAdmin");  
}
```

یعنی اگر session خالی باشد ورود امکان پذیر نیست و به صفحه دیگر انتقال می یابد.

فصل پنجم

جمع بندی و نتیجه گیری

نتیجه گیری

پروژه طراحی سایت خبری با هدف ارائه انواع اخبار متنی، تصویری، ویدیویی در انواع زمینه های سیاسی، اقتصادی، استانی، بین المللی، ورزشی، رسانه، دانش و فناوری و... طراحی شده به علاوه امکان حذف و اضافه و ویرایش انواع داده های درون سایت از قبیل گروه بندی خبر ها، برجسب های استفاده شده در سایت، مطالب خبری و رسانه های استفاده شده و... برای مدیر سایت در پنل مدیریت ارائه شده.

در پروژه سایت خبری سه عدد controller نوشته شده است:

- ❖ HomeController
- ❖ AdminController
- ❖ UserController

قسمت HomeController: مربوط به صفحه اصلی، ورود کاربران، ثبت نام کاربران و ورود مدیر میباشد.

قسمت AdminController: مربوط به صفحه خود مدیر، دیدن کاربران، دیدن خرید ها، دیدن مدیر های سایت، ویرایش مدیر ها و دیدن نظرات.

قسمت UserController: مربوط به صفحه خود کاربر، خرید و آپلود عکس، نمایش های خرید های قبلی، نمایش اطلاعات خود کاربر و گذاشتن نظر.

منابع و مراجع

<https://barnamenevisan.org> [۱]

[/https://www.daneshjooyar.com](https://www.daneshjooyar.com) [۲]

پیوست‌ها

جدول پ-۱: کد ویرایش admin در admincontoroller

```
[HttpPost]
public ActionResult EditAdmin(int id , string username , string password)
{
    SedaghatPro.Models.DomainModels.chapEntities db = new chapEntities();
    admin adm = db.admins.Where(a => a.id == id).SingleOrDefault();
    adm.username = username;
    adm.password = password;
    db.SaveChanges();
    return View(adm);
}

[HttpGet]
public ActionResult EditAdmin()
{
    chapEntities db = new chapEntities();
    string session = Session["username"].ToString();
    admin ad = db.admins.Where(a => a.username.Equals(session)).SingleOrDefault();
    return View(ad);
}
```

جدول پ-۲: کد ویرایش admin در قسمت view

```

@using (Html.BeginForm("EditAdmin", "Admin", FormMethod.Post))
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <h4>admin</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        @Html.HiddenFor(model => model.id)
        <div class="form-group">
            @Html.LabelFor(model => model.username, htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.username, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.username, "", new { @class =
"text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model => model.password, htmlAttributes: new { @class = "control-label
col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.password, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.password, "", new { @class = "text-
danger" })
            </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-2 col-md-10">
                <input type="submit" value="Save" class="btn btn-default" />
            </div>
        </div>
    </div>
}

```

جدول پ-۲: کد ویرایش Shop در قسمت UserController

```
[HttpGet]
public ActionResult Shop()
{
    return View();
}

[HttpPost]
public ActionResult Shop(print re, HttpPostedFileBase UploadImage)
{
    if (ModelState.IsValid)
    {
        re.image = UploadImage.FileName;
        string path = Server.MapPath("~/") + "Files\\UploadImages\\" +
        UploadImage.FileName;

        UploadImage.InputStream.ResizeImageByWidth(500, path,
        Utility.ImageComperssion.Normal);

        if (repo.AddPrint(re))
        {
            ViewBag.mess = "شد ثبت";
        }
        else
        {
            ViewBag.mess = "شد ثبت";
        }
    }
    else
    {
        ViewBag.mess = "است اشتیاه مقادیر";
    }
    return View();
}
```


جدول پ-۲: کد ویرایش Shop در قسمت view

```

@using (Html.BeginForm("Shop", "User", FormMethod.Post, new
{enctype="multipart/form-data", id = "myUploadForm" }))
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <h4>print</h4>
        <hr />
        @Html.ValidationSummary(true, "", new { @class = "text-danger" })
        <div class="form-group">
            @Html.LabelFor(model => model.idcustomer, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.idcustomer, new { htmlAttributes = new { @class = "form-control" } })
                @Html.ValidationMessageFor(model => model.idcustomer, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.nameproject, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model => model.nameproject, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model => model.nameproject, "", new { @class = "text-danger" })
            </div>
        </div>

        <div class="form-group">
            @Html.LabelFor(model => model.size, htmlAttributes: new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownListFor(Model => Model.size, new SelectListItem[] {
                    new SelectListItem() {Text = "a5", Value="a5"},
                    new SelectListItem() {Text = "a4", Value="a4"},
                    new SelectListItem() {Text = "a3", Value="a3"}})
            </div>
        </div>
    </div>
}

```

```
@Html.ValidationMessageFor(model => model.size, "", new { @class = "text-danger" })
</div>
</div>

<div class="form-group">
@Html.LabelFor(model => model.tedad, htmlAttributes: new { @class = "control-label
col-md-2" })
<div class="col-md-10">
@Html.DropDownListFor(Model => Model.tedad, new SelectListItem[]{
    new SelectListItem() {Text = "1000", Value="1000"},
    new SelectListItem() {Text = "2000", Value="2000"},
    new SelectListItem() {Text = "5000", Value="5000"}})
@Html.ValidationMessageFor(model => model.tedad, "", new { @class = "text-danger" })
</div>
</div>

<div class="form-group">
@Html.LabelFor(model => model.image, htmlAttributes: new { @class = "control-label
col-md-2" })
<div class="col-md-10">

    @Html.Upload("UploadImage")

@Html.ValidationMessageFor(model => model.image, "", new { @class = "text-danger" })
</div>
</div>
```