

مقدمه:

با توجه به اینکه در عصر ارتباطات به سر می بریم و تمامی انسان ها به نحوی تلاش می کنند تا با یگدیگر ارتباط برقرار کنند و هر یک از آن ها از روش های مختلفی برای برقراری ارتباط استفاده می کنند. یکی از روش های برقراری ارتباط گفتگو می باشد، که این روش به نوبه خود دارای انواع متفاوتی مانند گفتگو به صورت حضوری، گفتگو تلفنی، گفتگو با استفاده از اینترنت و تایپ کلمات و ... می باشد و هر یک از این روش های گفتگو دارای اهمیت خاص خود می باشد.

همانطور که گفته شد یکی از روش های برقراری ارتباط، گفتگو با استفاده از اینترنت و تایپ کلمات می باشد، که در این روش کاربران می توانند با استفاده از نرم افزارهای وب چت که بر روی اینترنت قرار دارند بادیگران در هر نقطه از جهان به گفتگو بپردازند. با توجه به صحبت های ذکر شده ما سعی نموده ایم تا در این پروژه نرم افزار وب چت را در اختیار کاربران عزیز قرار دهیم، و سعی نموده ایم تا با استفاده از فن آوری های جدید مانند آجاکس^۱ بر سرعت این نرم افزار بیفزاییم و مانع از اتلاف وقت کاربران عزیز شویم.

در این نرم افزار که در قالب یک سایت اینترنتی طراحی شده کاربران می توانند با عضویت در سایت از امکاناتی مانند: ارسال دعوت نامه دوستی برای دیگر کاربران، ارسال پیام متنی^۲، ارسال تصویر نما^۳، ارسال فایل^۴، ارسال نامه الکترونیکی^۵ و ... استفاده نمایند.

¹ Ajax
² Message
³ Icon
⁴ File Transfer
⁵ Email

فصل اول: ابزارها و نرم افزارهای مورد استفاده

۱-۱- ابزارها:

در این پروژه برای طراحی و پیاده سازی سایت از نرم افزارهای زیر استفاده شده است:

- آجاکس
- ASP.NET (C#)
- 'VS.NET 2005
- SQL Server 2005
- Photoshop 8
- وب سرور IIS^۲

¹ Visual Studio 2005

² Internet Information Service

۱-۲-۱ ASP.NET:

۱-۲-۱-۱ مقدمه ای بر ASP.NET:

ASP.NET نسل جدید ASP است که توسط شرکت مایکروسافت ارائه شده است. این محصول توسط مایکروسافت بعنوان شاخص اصلی فناوری در ساخت سایت های وب در نظر گرفته شده است. با استفاده از ASP.NET می توان هم اینترنت کوچک یک شرکت را ساخت و هم یک سایت وب تجاری خیلی بزرگ را طراحی و پیاده سازی نمود. مهمترین نکاتی که در طراحی این محصول در نظر گرفته شده است راحتی استفاده و بالا بودن کارایی و قابلیت آن می باشد.

۱-۲-۱-۲ ویژگی های ASP.NET:

- صفحات ASP.NET کامپایل می شوند.

هنگامی که یک صفحه ASP.NET برای اولین بار توسط یک مراجعه کننده به سایت فراخوانی می شود، آن صفحه ابتدا کامپایل شده و بر روی سرور نگه داشته می شود و در فراخوانی های بعدی از آن استفاده می شود. این بدین معنی است که صفحات ASP.NET خیلی سریع اجرا می شوند.

- صفحات ASP.NET با ابزارهای روی سرور ساخته می شوند.

با ابزارهای موجود در ASP.NET می توان صفحات پیچیده وب را براحتی طراحی نمود. بعنوان مثال با استفاده از ابزار دیتا گرید^۱ می توان به آسانی داده های موجود در یک بانک اطلاعاتی را تحت وب نمایش داد.

- مجموعه ASP.NET عضوی از بدنه NET است.

بدنه NET دارای بیش از ۴۵۰۰ کلاس آماده جهت استفاده در ASP.NET است. این کلاس ها تقریباً هر نیازی را در برنامه نویسی برآورده می کنند. بعنوان مثال از این کلاس ها می توان جهت تولید تصاویر بر حسب تقاضا، به رمز درآوردن یک فایل و یا ارسال یک نامه استفاده کرد.

۳-۲-۱- مقایسه ASP.NET و ASP کلاسیک:

^۱ DataGrid

ASP.NET نسل بعدی ASP کلاسیک است. اما این یک پیشرفت تکاملی است بطوریکه این دو فناوری تقریباً از یکدیگر متفاوتند. صفحات ASP با زبان‌های دستورالعمل نویسی مانند VBScript یا جاوااسکریپت^۱ ایجاد می‌شوند اما در ASP.NET ما یک فرایند کامل برنامه نویسی با زبانهای ویژوال بیسیک^۲ یا C# داریم. همچنین در ASP کلاسیک تنها پنج کلاس استاندارد (Request, Response, Application Session, Server) وجود دارد حال آنکه در ASP.NET می‌توان از بیش از ۴۵۰۰ کلاس استاندارد موجود در بدنه NET بهره جست. همچنین علیرغم قدرت و امکانات زیاد و متعدد ASP.NET، استفاده از آن در مقایسه با ASP کلاسیک بسیار آسانتر است. بعنوان مثال با استفاده از چند ابزار در یک صفحه ASP.NET می‌توان یک صفحه بسیار پیچیده HTML^۳ بدست آورد که ساخت آن در ASP کلاسیک ممکن است نیاز به چند روز کار داشته باشد.

۱-۲-۴- زبانهای برنامه نویسی در ASP.NET:

شما در ASP.NET می‌توانید از هر زبان برنامه نویسی که با بدنه NET سازگار باشد استفاده کنید. این زبان‌ها عبارتند از ویژوال بیسیک دات نت^۴ و C# و جاوا اسکریپت دات نت این بدین معنی است که شما جهت نوشتن برنامه در ASP.NET نیاز به فراگیری زبان جدیدی ندارید و اگر یکی از زبان‌های ویژوال بیسیک یا C++ یا جاوا را می‌دانید هم اکنون می‌توانید در ASP.NET برنامه بنویسید. از طرف دیگر تعدادی زبان‌های دیگر توسط بعضی از شرکت‌های فعال در این زمینه به مجموعه زبان‌های استاندارد ASP.NET افزوده شده‌است. بعنوان مثال اگر مایل باشید حتی می‌توانید از زبان پرل^۵ و کوبل^۶ هم در ASP.NET استفاده کنید.

۱-۲-۵- ابزارهای ASP.NET:

سالهاسست که برنامه نویسان ویژوال بیسیک جهت ساخت فرم‌های خود از ابزارهای ویژوال بیسیک مانند جعبه متن^۷ و جعبه لیست^۱ استفاده کرده‌اند. در ASP.NET هم شما می‌توانید از ابزارهای فراوان موجود در آن برای ساخت فرم‌ها و صفحات خود استفاده نمایید. در ASP.NET چهار دسته عمده از ابزارها موجود است:

^۱ JScript

^۲ Visual Basic

^۳ Hyper Text Markup Language

^۴ Visual Basic.Net

^۵ PERL

^۶ COBOL

^۷ TextBox

- ابزارهای اصلی مانند: جعبه متن، جعبه لیست، دکمه^۲
- ابزارهای اعتباری برای حصول اطمینان از ورود و تأیید صحت اطلاعات ورودی فرم‌ها.
- ابزارهای داده‌ای برای ارتباط با بانک اطلاعاتی و دستکاری داده.
- ابزارهای پیشرفته جهت نمایش عناصر پیچیده در واسط کاربر مانند تقویم و آگهی‌های تبلیغاتی.

۱-۲-۶- ساختار یک صفحه aspx:

صفحات ASP.NET دارای چندین بخش اصلی می‌باشند. این بخش‌ها را می‌توان در هفت مورد بررسی کرد:

- ۱- دایرکتیوها
- ۲- بلوکهای اعلان کد
- ۳- کنترل‌های ASP.NET
- ۴- بلوک‌های پردازش کد
- ۵- توضیحات طرف سرور
- ۶- دایرکتیو Include طرف سرور
- ۷- متن ساده و تگ‌های HTML

۱-۲-۶-۱- دایرکتیوها:

دایرکتیوها تنظیمات خاصی را برای صفحات aspx و ascx تنظیم می‌کنند که نحوه کامپایل شدن صفحه را مشخص می‌کند. دایرکتیوها با بلوک `<%@ %>` نشان داده می‌شوند و در هر کجای صفحه قابل تعریف می‌باشند معمولاً در ابتدای صفحه آورده می‌شوند.

در .NET Framework هشت نوع دایرکتیو وجود دارد که دو نوع Page و Import از بقیه پرکاربردتر هستند. دایرکتیو Page فقط در صفحات aspx قابل استفاده است و برخی خصوصیات Page نظیر زبان برنامه نویسی، نوع محتوا، نوع کد Page،

¹ ListBox

² Button

توضیحات مختصر در مورد صفحه، فعال بودن یا نبودن وضعیت ظاهر و... را مشخص می کند. مثلاً خط زیر به کامپایلر ASP.NET می گوید که زبان برنامه نویسی صفحه VB¹ و نوع محتوا، متن/XML است:

```
<%@ Page Language="VB" Content Type="text/xml" %>
```

دو خصوصیت دیگر Page، قابلیت ردگیری و قابلیت نمایش پیام های خطای زمان اجراست که برای اشکال زدایی از برنامه بکار می آیند. این دو خصوصیت بدین صورت فعال می شوند:

```
<%@ Page Trace="True" Debug="True" %>
```

دایرکتیو وارد کردن²، برای وارد کردن یک فضا نام³ به صفحه بکار می رود. به این ترتیب کلیه کلاس ها و واسطه های آن فضا نام در صفحه قابل استفاده می شود. این فضا نام می تواند جزئی از .NET Framework یا یک فضا نام تعریف شده توسط کاربر باشد. صورت کلی این دایرکتیو اینگونه است:

```
<%@ Import namespace="value" %>
```

۱-۲-۶-۲- بلوک های اعلان کد:

این بلوک ها برای تعریف متغیرها و متدها، زیربرنامه ها و توابع و به طور کلی برای تعریف منطق یک صفحه ASP.NET بکار می روند. شکل کلی یک بلوک اعلان کد چنین است:

```
<script runat="server" language="code language" Src="pathname">
```

Code goes here

```
</script>
```

تمامی کدهای یک صفحه ASP.NET در داخل چنین بلوکی قابل تعریف اند و اگر آنها را خارج این بلوک تعریف کنید، با پیغام خطا مواجه می شوید. خصوصیت دلخواه زبان⁴، زبان برنامه نویسی کد داخل بلوک را تعیین می کند. دقت کنید که اگر این خصوصیت تعیین نشود، زبان پیش فرض صفحه که توسط دایرکتیو Page مشخص شده، به عنوان زبان برنامه نویسی صفحه در نظر

¹ Visual Basic

² Import

³ NameSpace

⁴ Interface

⁵ Language

گرفته می شود. در ضمن زبان مشخص شده در تگ اسکریپت¹ با زبان مشخص شده در دایرکتیو Page باید یکسان باشند. در صورتی که هیچ زبانی انتخاب نشده باشد، زبان پیش فرض، ویژوال بیسیک است.

۱-۲-۶-۳- کنترل های ASP.NET :

کنترل های ASP.NET را می توان در کنار محتوای HTML بکار برد. تنها الزام در این مورد، قرار دادن آنها بین دو تگ `<form/>` و `<form runat="server">` است. البته در مورد بعضی تگ ها نظیر `` و `<asp:Label runat="server">` می توان این الزام را نادیده گرفت. یک محدودیت در این زمینه این است که فقط از یک تگ `<form runat="server">` می توان استفاده نمود. در واقع در صفحات ASP.NET امکان گروه بندی به چند فرم وجود ندارد.

۱-۲-۶-۴- بلوک های پردازش کد:

بلوک های پردازش کد برای اجرای کد موجود در محتوای متنی یا HTML یک صفحه ASP.NET بکار می روند. دو نوع بلوک پردازش کد وجود دارد که عبارت اند از: کد درون خطی و عبارت های درون خطی. کد درون خطی، دستور یا یکسری از دستورات را اجرا می کند. این کد با تگ های `<%>` و `<%= >` مشخص می شود. عبارت درون خطی، مقدار یک متغیر یا متد را نشان می دهد. عبارتهای درون خطی با کاراکترهای `<%>` و `<%= >` مشخص می شوند.

۱-۲-۶-۵- توضیحات طرف سرور:

ممکن است بخواهیم مانند تمام زبان های برنامه نویسی دیگر، مابین کدهای برنامه نویسی از توضیحاتی جهت مفهومی تر شدن کد، استفاده کنیم. در ASP.NET این توضیحات را در بین تگ های `<!--%>` و `<!--%=>` قرار می دهیم. در این صورت محتوای بین این تگ ها، چه کد ASP.NET باشد و چه متن ساده، مورد پردازش قرار نخواهد گرفت. یک نکته اینکه نمی توان چند توضیح طرف سرور را به صورت تودرتو استفاده کرد.

۱-۲-۶-۶- دایرکتیو Include طرف سرور:

با استفاده از این دایرکتیو می توان محتویات یک فایل مشخص شده را در هر کجای صفحه ASP.NET قرار داد. شکل کلی این دایرکتیو اینگونه است:

¹ Script Tag

```
<! -- #include file | virtual = filename -->
```

می توان هم از خصوصیت فایل¹ و هم از Virtual برای دادن مسیر فایل استفاده کرد. اگر از فایل استفاده کنیم، فایلی که می خواهیم شامل کنیم، نباید در دایرکتوری بالاتر از دایرکتوری فایلی که دستور Include در آن است باشد. اما در Virtual، مسیر فایل را می توان هر کجا که هست، به صورت کامل و نسبی وارد کرد. مثلاً اگر فایل ما در دایرکتوری به اسم MyDir که یکی از زیر شاخه های wwwroot است، باشد؛ باید دستور زیر را بکار برد:

```
<! -- #include virtual="/mydir/includefile.aspx" -->
```

دایرکتیو Include قبل از هر کدی در صفحه اجرا می شود، بنابراین برای تعیین مسیر فایلی که می خواهیم شامل کنیم نباید از متغیرها استفاده کنیم.

۱-۲-۶-۷- متن ساده و تگ های HTML:

آخرین ساختاری که می توان در یک صفحه ASP.NET بکار برد، محتوای HTML است. قسمت های ثابت صفحه، با همان تگ های معمولی HTML و متن ساده ساخته می شود. در واقع محتوای HTML صفحه ASP.NET همراه با بقیه کدها کامپایل می شود.

۱-۲-۷- معماری فایل ASP.NET:

در عمل و ساختار برنامه های کاربردی ASP.NET از فایل های زیادی استفاده می شود.

✓ **.asax**

این پسوند برای فایل خاصی بنام Global.asax استفاده می گردد. این نوع فایل محتوی ترکیب نحوی رویداد برای نوشتن رویدادهای سطح برنامه کاربردی ASP.NET است. آن را می توان در دایرکتوری ریشه یک برنامه کاربردی ASP.NET پیدا کرد.

¹ File

✓ .ascx

این نوع فایل نمایانگر یک کنترل تعریف شده توسط برنامه نویس ASP.NET است. کنترل‌های کاربر¹ در صفحات ASP.NET عموماً از کنترل‌های سرویس دهنده (کادرهای متنی، کادرهای فهرست، دکمه‌ها و غیره) تشکیل شده‌اند که عناصر اولیه صفحه وب را می‌سازند. ASP.NET درست مانند برنامه‌سازی مرسوم و ویژوال بیسیک امکان ایجاد کنترل‌ها-ی تعریف شده توسط برنامه‌نویس را فراهم می‌کند. این کنترل‌ها معمولاً از ترکیب کنترل‌های سرویس دهنده و برنامه-سازی برای انجام یک وظیفه یا مجموعه‌ای از وظایف بخصوص تشکیل می‌شوند.

✓ .asmx

این پسوند برای سرویس‌های XML وب استفاده می‌گردد. این نوع فایل از سوی آن دسته از سرویس‌های وب میزبان مورد استفاده قرار می‌گیرد که از راه دور یا بطور محلی در معرض برنامه‌های کاربردی NET هستند. سرویس وب موجودیتی قابل برنامه‌ریزی است که برای برنامه کاربردی یک عنصر خاص کارکرد معینی را تأمین می‌کند.

✓ .aspx

از این پسوند که اصلی‌ترین پسوند ASP.NET است برای فرم‌های وب و صفحات معمولی ASP.NET استفاده می‌گردد.

✓ .axd

این نوع فایل مربوط به پیگیری² برنامه کاربردی ASP.NET است و به ASP.NET اجازه می‌دهد تا به جمع‌آوری اطلاعات درباره درخواست‌های HTTP برای یک برنامه کاربردی بپردازد.

✓ .vsdisco

این نوع فایل XML، لینک‌ها را در معرض منابع دیگری قرار می‌دهد که سرویس وب را توصیف می‌کنند. از فایل VSDISCO برای کشف سرویس‌های وب استفاده می‌شود که بطور عمومی موجود هستند.

¹ User Control

² Tracing

.htm ✓

فایل‌های معمولی از نوع HTML را در برمی‌گیرند.

.xml ✓

این نوع سند XML مخصوص استفاده در برنامه‌های کاربردی ASP.NET است. این فایل XML برای مقاصد مختلفی استفاده می‌شود از جمله نگهداری اطلاعات برنامه کاربردی و نیز مجموعه داده‌های بازگشتی از بانک اطلاعاتی.

.vb ✓

این فایل شامل کد ویژوال بیسیک است که آن نیز به نوبه خود به یک فایل ASPX یا ASCX به ارث می‌رسد و کلیه عملیات برنامه‌ای مربوط به صفحات ASP.NET در آن قرار می‌گیرد. این نوع فایل اصطلاحاً کد پنهان^۱ نیز نامیده می‌شود.

.cs ✓

همانند پسوند VB است اما بجای کد ویژوال بیسیک حاوی کد زبان C# است.

.config ✓

پسوند فایل web.Config است. این نوع فایل بر یک فایل پیکربندی دلالت دارد یعنی فایلی که از آن برای تعیین صفات مشخصه مختلف برنامه کاربردی استفاده می‌شود. این صفات مشخصه شامل تنظیمات اشکال زدایی^۲، تأیید امنیت^۳، کارکرد پیگیری^۴، مدیریت جلسه کاری^۵ و عمومی سازی^۶ می‌شوند.

۱-۲-۸- شیء جلسه^۷ در ASP.NET:

¹ Code-behind
² Debugging
³ Authentication
⁴ Tracing
⁵ State management
⁶ Globalization
⁷ Session

ASP.NET تنظیمات مدیریت وضعیت پیشرفته‌ای نسبت به شیء جلسه در ASP کلاسیک دارد. این تنظیمات در فایل Web.Config پی‌کرنندی می‌شوند.

پی‌کرنندی جلسه:

وضعیت جلسه می‌تواند در قسمت <Session State> فایل Web.Config پی‌کرنندی شود.

رویدادهای جلسه:

دو رویداد مربوط به شروع و خاتمه جلسات در فایل Global.aspx وجود دارند که می‌توانید آنها را اداره کنید. این رویدادها شروع جلسه^۱ و پایان جلسه^۲ نام دارند. مثلاً به کمک آنها می‌توان تعداد کاربران آنلاین وب سایت را در صفحه نشان داد. برای این کار باید در رویداد شروع جلسه یک واحد به تعداد کاربران آنلاین اضافه کنیم و یک واحد از این تعداد را در رویداد پایان جلسه کم کنیم.

استفاده از شیء جلسه:

شیء جلسه در واقع یک خصوصیت کلاس Page^۳ و همچنین یک نمونه از کلاس HttpSessionState است که در فضا نام System.Web.SessionState تعریف شده است. استفاده از شیء جلسه همانند استفاده از یک مجموعه^۴ است.

۱-۲-۹- فضانام‌ها در ASP.NET:

یک نکته مهم که در زمان استفاده از .NET Framework باید به آن توجه داشت آن است که فضانام در ساختمان برنامه کاربردی قرار دارند. فضانام یک طرح نامگذاری منطقی برای گروه بندی کلاس‌های مرتبط است. این طرح مانع از آن می‌شود تا کلاس‌هایی که برای متدها و خصوصیات از یک شناسه یکسان استفاده می‌کنند تداخل داشته باشند. مثلاً .NET Framework برای گروه بندی تایپ‌ها به مقوله‌های منطقی عملکرد، از قبیل چارچوب برنامه کاربردی ASP.NET، از یک طرح نامگذاری سلسله مراتبی استفاده می‌کند. ابزارهای طراحی از فضانام‌ها با هدف تسهیل مرور و ارجاع تایپ‌ها در برنامه بهره برداری می‌کنند. می‌توانید در فضانام‌های خود از یک ساختار سلسله مراتبی استفاده کنید. قرار دادن اشیاء

^۱ Session_Start

^۲ Session_End

^۳ Page

^۴ Collection

مشابه تحت زیرعنوانها در یک فضای مشترک تشخیص هدف فضای را آسانتر می کند، و در عین حال باعث می شود برنامه به مراتب شیء گراتر شود.

فضانامها در ساخت برنامه های کاربردی ASP.NET نقش مهمی ایفا می کنند. خوشبختانه لازم نیست برای همه اشیا که می توانند به وسیله صفحات ASP.NET مورد استفاده قرار بگیرند سیستم طبقه بندی فضای ایجاد کنید. مایکروسافت این مساله را برای شما حل کرده است. دو فضای ریشه، و فضای فرزند آنها را می توان وارد صفحات ASP.NET خود کرد. اولی سیستم نامیده می شود، و دومی مایکروسافت نام دارد.

• فضای سیستم^۱:

فضانام سیستم فضای اصلی برای ساخت ASP.NET و همه برنامه های کاربردی دیگر مبتنی بر .NET Framework است. هر چیزی که در برنامه کاربردی شما قابل انجام باشد از طریق فضای سیستم کنترل می شود. ۹ فضای پیش فرض (فضانام سیستم و ۸ فرزند آن) وجود دارند که به صورت خودکار به صفحات ASP.NET اضافه می شوند:

- System
- System.ComponentModel.Design
- System.Data
- System.Drawing
- System.Web.SessionState
- System.Web
- System.Web.UI
- System.Web.UI.WebControls
- System.Web.UI.HtmlControls

• فضای مایکروسافت^۲:

علاوه بر فضای سیستم که در چارچوب .NET یافت می شود، مایکروسافت چند فضای اضافه کرده است که برای زبان برنامه سازی ای که می خواهید از آن در برنامه کاربردی خود استفاده کنید عملکرد لازم را تامین می کنند. ممکن است شما بصورت مستقیم با این فضای کاری نداشته باشید.

¹ System Namespace

² Microsoft Namespace

استفاده از فضاها در صفحات ASP.NET:

دو راه برای افزودن فضاها به برنامه کاربردی ASP.NET وجود دارد. از شبهه دستور `@Import` برای صفحات ASPX و از کلمه کلیدی `Imports` برای افزودن فضاها به برنامه مربوطه در ویژوال بیسیک استفاده می‌شود و برای زبان C# از دستور `using` استفاده می‌گردد. قطعه برنامه زیر ترکیب نحوی برای افزودن فضاها `System.Web.UI.WebControls` به صفحه ASP.NET شما است.

```
<%@ Import namespace = "System.Web.UI.WebControls" %>
```

همین فضاها را در قسمت کد برنامه بصورت زیر به برنامه اضافه می‌کنیم.

```
Imports System.Web.UI.WebControls (vb.net)
```

```
Using System.Web.UI.WebControls; (C#)
```

در صورتی که می‌خواهید چند فضاها را به صفحه ASP.NET خود و یا صفحه کد اضافه کنید باید هر کدام را جداگانه اضافه کنید. بعنوان مثال، برای افزودن فضاها `System.Web.UI.HtmlControls` به صفحات با فضاها موجود، درست بعد از آخرین عبارت مهم به خط بعد بروید و `Imports System.Web.UI.HtmlControls` را اضافه کنید. به محض آنکه `Imports System` را تایپ کنید، `VS.NET` فهرستی از فضاها را ظاهر می‌کند، و می‌توانید به سادگی فضاها مورد نظر را با ماوس برگزینید.

۱-۲-۱۰- فایل web.config:

بیشتر برنامه‌های کاربردی کامپیوتری دارای بخشی به نام نصب^۱ هستند که برای انجام تنظیمات متناسب با استفاده‌های مختلف کاربران مورد استفاده قرار می‌گیرد. برنامه‌های کاربردی مبتنی بر وب نیز که دارای کاربران زیادی هستند که از این قائده مستثنی نیستند و احتیاج به انجام تنظیماتی متناسب با کاربردهای مختلف خود دارند.

در ASP.NET اینگونه تنظیمات که مربوط به منابع مورد استفاده برنامه کاربردی، اطلاعات محلی، اطلاعات امنیتی و... می‌شوند در داخل یک فایل XML ذخیره می‌شوند که نام آن `Web.Config` است. این فایل شامل تعداد زیادی تگ با خصوصیات مختلف است که هر یک به جنبه‌ای از یک برنامه کاربردی تحت وب مربوط هستند.

¹ Visual Studio.net

² SetUp

تنظیمات هر فایل Web.Config به فایل‌های داخل زیر دایرکتوری‌ای که در آن قرار دارد و همچنین زیر دایرکتوری‌های آن اعمال می‌شود. همچنین برنامه‌نویس می‌تواند از طریق کدهای داخل هر فایل Web.Config منابع و دایرکتوری‌های مورد نظر را تغییر دهد.

۱-۲-۱۱- بهینه‌سازی و افزایش کارایی و سرعت برنامه‌های ASP.NET:

اگرچه ASP.NET مدل جدیدی از برنامه‌نویسی تحت وب می‌باشد و آمده‌است که نواقص و کمبودهای قبلی را برطرف نموده و به کارآیی برنامه‌های تحت وب بیافزاید، اما شما همچنان نیاز دارید که جهت کارآیی و بهینه‌بودن برنامه‌هایتان نکاتی را رعایت کنید. این نکات جهت افزایش کارآیی، سرعت و صرفه‌جویی در مصرف منابع بر روی سرور است.

❖ حالت جلسه^۱ را در صورت عدم نیاز حتما غیرفعال نمائید:

بصورت پیش فرض وضعیت جلسه برای هر صفحه فعال است. اگر مطمئن هستید که در آن صفحه از متغیرهای جلسه استفاده نمی‌کنید، حتما آن را در دایرکتیو Page^۲ بالای صفحه تان با قراردادن آن برابر نادرست^۳ غیرفعال کنید تا بدین ترتیب منابع اضافی در سرور مصرف نشود.

❖ از کنترل‌های سرور با احتیاط استفاده کنید:

اگرچه کنترل‌های سرور دارای قابلیت‌های بسیار زیاد و جالبی می‌باشند اما این قابلیت‌ها ناشی از اجرای آنها بر روی سرور است. از آنجا که هر اجرائی بر روی سرور ایجاد بار اضافی در سرور می‌کند لذا در استفاده از این نوع کنترل‌ها خست بخرج دهید و اگر جایی می‌توانید از کنترل‌های معمولی HTML استفاده نمائید حتما این کار را بکنید.

^۱ Session State

^۲ Page

^۳ False

❖ از Page.IsPostBack استفاده کنید.

استفاده از Page.IsPostBack سبب کاهش رفت و برگشت‌های^۱ اضافی می‌گردد. همچنین به شما کمک می‌کند که تشخیص دهید چه موقع نیاز به تولید داده جدید برای یک صفحه دارید.

❖ تا آنجائیکه ممکن است از حالت دید^۲ استفاده نکنید.

حالت دید هم مانند حالت جلسه برای نگهداری اطلاعات و سابقه یک فرم از منابع سرور استفاده کرده و به حجم صفحات اضافه می‌کند. از طرفی این حالت بصورت پیش فرض فعال می‌باشد. لذا در صورتی که به آن نیاز ندارید از آن استفاده نکنید و حالت آن را غیرفعال نمایید.

❖ از روال‌های ذخیره شده^۳ استفاده کنید.

اگر از بانک اطلاعاتی در برنامه‌تان استفاده می‌کنید، سعی کنید که حتما پرس و جوهای^۴ خود را به روال‌های ذخیره شده بسپارید. از آنجائیکه این روال‌ها برای مراجعه اول کامپایل شده و از آن پس جهت اجرا در اختیار مدیر بانک اطلاعاتی^۵ قرار می‌گیرد، نسبت به پرس و جوهای معمولی درون برنامه ای بسیار کارا تر و سریعتر هستند. از طرفی با بهینه سازی خاصی که برای گرداننده^۶ SQLServer در مجموعه NET شده‌است، سرعت اجرای روال‌های ذخیره شده نسبت به روش‌های معمولی OLEDB یا ODBC بیش از دو الی سه برابر می‌باشد.

❖ از خواننده داده^۷ استفاده کنید.

اگر حجم بالایی از داده ندارید یا تنها می‌خواهید از بانک اطلاعاتی عمل خواندن را انجام دهید حتما از خواننده داده استفاده کنید. استفاده از دیتاست^۸ سبب کندی کار و مصرف بالای منابع می‌گردد.

¹ Round Trip
² View State
³ Stored Procedure
⁴ Query
⁵ DBMS
⁶ Driver
⁷ Data Reader
⁸ DataSet

۱-۲-۱۲- ذخیره کردن فایل سمت سرور در ASP.NET:

کدهای مربوط به ذخیره کردن فایل سمت سرور را باید به دو قسمت تقسیم نمود. قسمت اول شامل کدهایی است که نحوه نمایش در مرورگر را شامل می‌شوند و اساس آن بر HTML است. و قسمت دوم مربوط به طرف سرور است که شامل کدهایی است که با ASP.NET نوشته می‌شوند.

کدهای طرف مشتری^۱:

به دلیل آنکه در این قسمت، ارسال اطلاعات به سرور را داریم، باید کدهای مربوطه درون یک فرم قرار گیرند. اما لازم است که ویژگی `encType` آن فرم را به صورت زیر مقداردهی کنیم، زیرا در غیر اینصورت فایل انتخاب شده توسط کاربر به سرور ارسال نخواهد شد:

```
encType="multipart/form-data"
```

در HTML کنترل وجود دارد که وظیفه دریافت نام فایل و ارسال آن به سرور را عهده دار می‌باشد:

```
<input id="MyInputFile" type="file" name="MyInputFile" runat="server">
```

توجه کنید که این کنترل را به صورت یک سرور کنترل در می‌آوریم تا بتوانیم در ASP.NET از توانایی‌های آن استفاده نماییم و به همین دلیل ویژگی `runat="server"` را به کنترل مربوطه اضافه کرده ایم. بنابراین کد مربوط به قسمت مشتری به صورت زیر خواهد بود:

```
<body>  
<form method="post" encType="multipart/form-data">  
<INPUT id="MyInputFile" type="file" name="MyInputFile" runat="server">  
<input type="submit" value="Submit">  
</form>  
</body>
```

کدهای طرف سرور:

¹ Client

این قسمت کدهایی را شامل می‌شود که برای ذخیره کردن فایل بر روی سرور دهنده مورد استفاده قرار می‌گیرند. همانطور که گفته شد شیء مربوط به دریافت نام فایل را به صورت سرور کنترل قرار دادیم تا بتوانیم از خصوصیات آن در سرور استفاده کنیم (در اینجا نام آنرا MyInputFile در نظر گرفته ایم). یکی از خصوصیات این شیء، شیء Posted File می‌باشد که به فایل ارسالی توسط کاربر اشاره می‌کند. این شیء نیز شامل متدی است که اقدام به ذخیره فایل می‌کند. بنابراین قبل از ذخیره کردن باید مطمئن شد که آیا شیء Posted File به چیزی اشاره می‌کند و یا اینکه تهی است. برای این منظور کد مربوط به ذخیره سازی را در شرط زیر قرار می‌دهیم:

If Not (MyInputFile.PostedFile Is Nothing) Then

و در صورت صحیح بودن شرط، اقدام به ذخیره سازی می‌نماییم. متد Save As شیء Posted File دارای پارامتری است که محل و نام فایلی که قرار است بر روی سرور ذخیره شود را مشخص می‌کند. در انتها نیز پیغامی مبنی بر اینکه فایل با موفقیت ذخیره شده است را به کاربر می‌دهیم.

Response.Write ("Your File was saved on the server successfully...")

بنابراین کد طرف سرور به صورت زیر خواهد بود:

```
<script language="vb" runat="server">  
sub page_load(s as object, e as EventArgs)  
If Not (MyInputFile.PostedFile Is Nothing) Then  
MyInputFile.PostedFile.SaveAs ("c:\filename.ext")  
Response. Write ("Your File was saved on the server...")  
End If  
end sub  
</script>
```

۱-۳-۳: Sql Server

۱-۳-۱-۱- مقدمه ای بر sql server

SQL Server 2000 یک موتور^۱ پایگاه داده ارائه شده توسط شرکت مایکروسافت است و ادامه ویرایش SQL Server 7 است. بدون اقرار می توان گفت این محصول پر کاربرد ترین بسته نرم افزاری^۲ پایگاه داده محسوب می شود. البته مایکروسافت ادعا کرده است که بیش از ۶۰٪ بازار پایگاه داده^۳ جهان را به خود اختصاص داده و این روز به روز در حال افزایش است. شرکت IBM در اواخر دهه ۷۰ زبان SEQUEL که برای پرس و جوها^۴ استفاده می شد را ابداع کرد. با تغییر زمان این زبان نیز تغییراتی کرد و تا آنجا که فقط برای پرس وجوها استفاده نمی شد و از آن می توانستیم برای مدیریت^۵ و امنیت^۶ در پایگاه داده نیز استفاده کنیم و بعد از مدتها این زبان به SQL تغییر نام داد.

امروزه ورژنهای متعددی از این نرم افزار استفاده می شود که نرم افزار Microsoft SQL Server 2000 از Transact SQL که به آن TSQL هم می گویند استفاده می کند. SQL Server 2000 هم اکنون دارای ۲ نسخه است: یکی SQL 2000 Personal Edition Server که بر روی مشتری^۷ نصب می شود (- Windows ME - Windows 98 Professional Edition - Windows XP Home Edition - Windows XP Professional Edition) البته اگر بر روی Windows 98 و Windows ME نصب کنیم، بسیاری از قابلیت های این بسته نرم افزاری از بین می رود. بهترین حالت نصب این نسخه بر روی سیستم عامل های Windows XP و Windows Professional 2000 است.

یک نسخه دیگر این نرم افزار: Enterprise Edition SQL Server 2000 است که مختص سرورها^۸ است یعنی فقط بر روی تمامی نگارش های Window Server 2K و Windows Server 2003 نصب می شود. اغلب برنامه نویسان بانک های اطلاعاتی، برای برقراری ارتباط بین محیط کاربر^۹ و بانک اطلاعاتی مجبور به استفاده از دستورات SQL هستند. در هنگام استفاده از این دستورات همواره سرعت و کارایی مهمترین پارامتر به حساب می آیند.

۱-۳-۲- خلاصه ای از دستورات SQL:

¹ Engine
² Package
³ Database
⁴ Query
⁵ Administrating
⁶ Security
⁷ Client
⁸ Server
⁹ user interface

- دستور Select:

این دستور که دستوری مستقل نیست و حتما باید با اجزایی بکار رود جهت ساخت پرس و جو بر روی بانک اطلاعاتی بکار می‌رود و رکوردهایی که با شرایط این دستور همخوان باشد به عنوان نتیجه پرس و جو برمی‌گرداند. چهار کلمه کلیدی وجود دارند که بخش‌های ارزشمند این دستور را تشکیل می‌دهند:

select -۱

from -۲

where -۳

order by -۴

شکل کلی دستور:

Select [*|distinct column1, column2...]

From [table, table2...]

Where شرط

Order by نام فیلد یا شماره فیلد

استفاده از توابع در Select:

۱- Count: تعداد سطرهای بازگردانده شده توسط select را می‌شمارد.

Select Count (*) from Customers

Where Country='uk'

در اصل تعداد مشتریانی را می‌شمارد که در کشور انگلیس هستند. که عدد ۷ جواب است.

۲- Sum: مجموع یک فیلد عددی را برمی‌گرداند.

Select sum (Quantity) from [Order Details]

استفاده از group by:

هنگامی که از توابع count و Sum به همراه یک فیلد دیگر در دستور select استفاده می‌شود از group by استفاده

می‌کنیم.

نکته: دستور where می تواند خود شامل یک دستور select باشد. تنها نکته ای که می بایست توجه کرد این است که نام فیلدی که در شرط آورده می شود حتما در دستور select آورده شود، به عبارت دیگر select درون شرط تنها یک ستون را می بایست برگرداند.

۳- Min, max: بیشترین و کمترین مقدار فیلد را در بانک اطلاعاتی بدست می دهد.

```
Select min (Quantity)
From [Order Details]
```

۴- Top n: تعداد n سطر اول بانک اطلاعاتی را برمی گرداند.

۵- Into

```
Select * from table1 into table2
```

اطلاعات table1 را به table2 کپی می کند. البته table2 باید از قبل وجود نداشته باشد. این دستور خود table2 را می سازد. دستور select قویترین و کاربردی ترین دستور در SQL است که خود ماهها نیازمند تمرین و آموزش است.

- دستور Delete:

برای حذف اطلاعات از یک بانک اطلاعاتی استفاده می شود.

شکل کلی دستور:

```
Delete table1
Where شرط
```

- دستور insert:

برای اضافه کردن اطلاعات به یک جدول از این دستور استفاده می شود.

```
Insert into table1 (f1, f2...)
Values (v1, v2...)
```

که f1, f2 نام فیلدها و v1, v2 مقادیر آنها می باشد. البته می توانید مقادیر را نتیجه یک select قرار دهید.

۱-۳-۳- Enterprise Manager

می توان گفت قلب SQL Server است. در Enterprise Manager شما می توانید یک اتصال به سرور SQL برقرار کنید. جدول بسازید، کاربر^۱ تعریف کنید و... علامت فلش سبز نمایانگر فعال بودن سرور است. سرور می تواند محلی^۲ باشد مانند بالا و یا یک SQL Server باشد بر روی یک سرور. برای ایجاد یک سرور جدید یا به عبارت دیگر وصل شدن مشتری (ویندوز ۹۸) به یک سرور دیگر بر روی یکی از Microsoft SQL Server یا SQL Server Group و یا بر روی سرور موجود کلیک سمت راست کرده و گزینه New SQL Server Registration را انتخاب کنید. سپس کلید next را انتخاب کنید. سپس در منوی بعدی در available Server نام سرور خود را تایپ کنید. (نام سرور SQL خود را که بر روی ویندوز ۲۰۰۰ خود نصب کرده اید) و بعد کلید add را فشار دهید و گزینه next را انتخاب کنید.

در پنجره بعد از شما سوال می کند که آیا از نام کاربری^۳ ویندوز استفاده کند و یا اینکه از یک نام کاربری مخصوص خود SQL Server استفاده کنید. گزینه دوم را انتخاب کرده و سپس Login Name و Password را وارد کنید. (در حالت پیش فرض sa بعنوان login name و فضای خالی به جای Password وارد کنید). پس از چند بار فشار کلید next شروع به وصل شدن به Server می کند. در صورت موفقیت آمیز بودن با پیامی این کار را اطلاع می دهد.

از میان دیگر گزینه ها شما بیشترین استفاده را از پایگاه داده^۴ خواهید کرد. به عبارت دیگر هر کار و پروژه ما بعنوان یک پایگاه داده در سرور SQL قرار می گیرد. همه جداول و دستورات مربوط به آنها در این محل نگهداری می شود. با کلیک سمت راست بر روی Databases و انتخاب New Database می توانید یک پایگاه داده جدید برای خودتان بسازید. پس از انتخاب نام آن را تایید کنید.

۱-۳-۴- اجزای Database:

هر Database شامل موارد زیر است:

۱- Diagram: ارتباط جداول را نشان می دهد.

۲- Tables: جداول پروژه را نشان می دهد.

۳- Views: دیدهای پروژه را نشان می دهد.

^۱ User

^۲ Local

^۳ Username

^۴ Database

۴- Stored Procedure: کدهای SQL مربوط به عملیات روی جداول را نگهداری می کند.

۵- Users: کاربران تعریف شده بر روی این database را نشان می دهد.

۶- Roles: قوانین دسترسی به جداول و پروسیجرها را نشان می دهد.

۷- Rules: قوانین مربوط به داده ها را در جداول نشان می دهد.

گزینه های ۷ به بعد کاربرد آنچنانی برای کارهای ابتدایی ندارند.

۱-۳-۵- ایجاد یک جدول جدید:

برای ایجاد یک جدول جدید بروی tables کلیک سمت راست کرده و گزینه New Table را انتخاب کنید. سپس در کادر بعدی نام جدول را انتخاب کنید. حال فیلدها و نوع آنها را مشخص کنید. بعد از مشخص کردن نوع و احتمالاً طول فیلد را باید مشخص کنید که آیا فیلد همیشه می بایست مقدار داشته باشد و یا می تواند تهی^۱ باشد. Allow Nulls اگر تیک داشته باشد یعنی این فیلد می تواند هیچ مقداری به خود اختصاص ندهد.

تذکر: مقدار تهی را با فضای خالی اشتباه نگیرید.

در قسمت مقدار پیش فرض^۲ مقدار اولیه برای فیلد را وارد کنید تا در صورتی که هیچ مقداری درج نشد آن مقدار درج شود.

اگر Identity را تیک بزنید این فیلد بشکل خود افزا خواهد شد که اولاً باید نوع فیلد عددی و ثانياً مقدار گامها در Identity increment مشخص شود. مقدار اولیه آن را هم می توانید در Identity Seed قرار دهید. بدین شکل با این مقدار شروع و با گامهای مشخص شده حرکت خواهد کرد.

تذکر: هیچ مقداری در این فیلد نمی توانید درج کنید، چراکه خود سیستم این مقدار را تولید می کند.

برای مشخص کردن فیلد کلید (یا فیلدهای کلیدی) فیلد(ها) را انتخاب و بر روی علامت کلید بر روی میله ابزار^۳ کلیک کنید.

برای تعریف ایندکس^۴ کلیک سمت راست کرده و index/keys را انتخاب کنید.

۱-۳-۶- ایجاد روال های ذخیره شده:

¹ Null

² Default Value

³ Toolbar

⁴ Index

مانند هر زبان دیگر رویه‌ها در SQL Server نیز موجود می‌باشند و بکار می‌روند. سه رویه Delete , Update , insert را می‌توانید برحسب نیاز با ویزاردهای خود SQL Server بسازید. از اینجا به بعد از واژه SP بجای رویه استفاده خواهیم کرد. برای ایجاد یک SP جدید ابتدا بر روی گزینه Stored Procedure کلیک سمت راست کرده و گزینه new Stored Procedure را انتخاب کنید. در پنجره بعدی شما می‌توانید متن SP را وارد کنید.

۱- نام sp: ابتدا بجای [PROCEDURE NAME] یک نام برای SP خود در نظر بگیرید. من خود از ساختار زیر بدین منظور استفاده می‌کنیم: عملیات_نام جدول
به عنوان مثال اگر نام جدول Sale و عملیات مورد نظر یک عملیات insert باشد نام SP را Sale_INSERT می‌گذاریم. بهتر است نام عملیات با حروف بزرگ تایپ شود. البته بعضی از دوستان از سه حرف عملیات استفاده می‌کنند. برای مثال بالا خواهیم داشت: Sale_INS.

۲- تعریف پارامترها: برای تعریف پارامترهای ورودی SP قبل از کلمه کلیدی As آنها را داخل پرانتز مشخص کنید. بدین شکل که ابتدا علامت @ سپس نام پارامتر بعد فاصله و نوع پارامتر.
تذکر: تمامی متغیرها در SP از ساختار نام متغیر @ پیروی می‌کنند.

۳- تعریف متغیرها: برای تعریف متغیرها می‌بایست از کلمه کلیدی Declare استفاده کنیم. بعنوان مثال:

```
Declare @myname varchar (50)
```

متغیر @myname از نوع کارکتری پویا تعریف می‌کند.

۴- برگرداندن کد خطا: بدین منظور از دستور @Error return استفاده می‌کنیم.

گرچه هر مقداری را که بخواهیم می‌توانیم با دستور return برگردانیم.

۵- تعریف پارامترها که مقداری را برمی‌گردانند: برای این منظور هنگامی که پارامتر را در ابتدای پروسیجر تعریف می‌کنیم بعد از

نوع آن از کلمه کلیدی output می‌کنیم بعنوان مثال:

```
Create Stored Procedure Employees_BROWSE2
```

```
(EndDate char(10) , @pp varchar(10) @output ,@StartDate char(10) )
```

```
As
```

۶- مقدار دهی به متغیرها: به دو روش می‌توانید این کار را بکنید یکی با دستور Set و دیگری با دستور Select.

۷- دستورات شرطی: مانند دیگر زبان‌ها شما می‌توانید در SQL دستورات شرطی را بکار ببرید. ساختار آن به شکل زیر است:

دستور then شرط if

Then شرط if

Begin

دستور ۱

دستور ۲

...

End

۸- دستور `Set NOCOUNT on`: این دستور از نوشتن تعداد سطرهای برگردانده شده توسط دستورات جلوگیری می‌کند.

کاربرد مهم آن زمانی است که شما چند کار را پشت سرهم در SP انجام می‌دهید. مثلاً یک جدولی موقت می‌سازید و پس از آن یک تعدادی از فیلدها را با دستور `Select` انتخاب می‌کنید. اگر این دستور را در ابتدای SP استفاده نکنید، هنگامی که می‌خواهید از آن SP در یک زبان برنامه‌نویسی استفاده کنید با خطای SP هیچ دیتا ستی^۱ برنمی‌گرداند مواجه خواهید شد.

تذکر: شما می‌توانید یک SP را در یک SP دیگر فراخوانی کنید. برای این منظور همانطور که قبلاً گفته شد از دستور `exec` استفاده نمایید.

تذکر: با دستور `exec` شما می‌توانید یک دستور SQL را نیز اجرا کنید. این کار زمانی بکار می‌آید که دستور مورد نظر پویا و متغیر باشد.

دید^۲:

دید در اصل یک جدول مجازی است که محتوی آن توسط یک پرس‌وجو تعریف می‌گردد. همانند جدول دید هم دارای

سطر و ستون‌هایی می‌باشد. می‌توان به موارد زیر به عنوان مزایای دید اشاره کرد:

^۱ Dataset

^۲ View

دیدها به کاربران اجازه می دهند تا بر روی داده‌هایی که نیاز دارند متمرکز شوند. بنابر این داده‌های غیر ضروری می توان از دید خارج کرد. دیدها امنیت داده‌ها را نیز افزایش می دهند چراکه کاربر فقط داده‌هایی را می بیند که در دید وجود دارند. دیدها به کاربران اجازه می دهند تا داده‌ها را به روش‌های متفاوت مشاهده نمایند. دیدها می توانند برای مبادله داده‌ها با سایر برنامه‌های کاربردی بکار روند.

ایجاد دید:

وقتی دیدی را ایجاد می کنید نام آن می بایست در بین نام جداول و دیگر دیدهایی که کاربر مورد نظر آنها را ساخته، یکتا باشد. در SQL Server 7.0 شما امکان ایجاد شاخص بر روی دیدها را نداشتید، لیکن این امکان در SQL 2000 اضافه شده است.

مراحل ایجاد یک دید:

۱- Manager expires را باز کرده و بر روی Databases کلیک کرده و پایگاه داده ای را که می خواهید دید در آن ایجاد کنید. باز کنید.

۲- روی Views کلیک راست کرده و سپس گزینه New View را انتخاب کنید.

۳- در پنجره بعدی کلیک سمت راست کرده و گزینه Add Table را انتخاب کنید.

۴- دکمه‌های Table و یا Views جدول و یا دیدهای مورد نظر را انتخاب نمایید. و بر روی دکمه Add کلیک نمایید. این کار را برای تمام جداول و یا دیدهای مورد نظر تکرار کنید و سپس بر روی دکمه Close کلیک نمایید.

۵- در قسمت Column از پانل مشبک، ستون‌هایی را که می خواهید در دید به آنها ارجاع نمایید انتخاب کنید. اگر می خواهید ستونی در مجموعه نتیجه دیده شود گزینه Output متناظر با آن می بایستی حتما تیک داشته باشد.

۶- در ستون Criteria شرط را بنویسید. چند شرط را می توانید در ستون‌های OR تکرار کنید. برای گروه بندی بر روی ستون Criteria کلیک سمت راست کرده و گزینه Group By را انتخاب نمایید. در اینصورت شما قادر خواهید بود از توابعی همچون Sum استفاده کنید.

تذکر: اگر Group by را انتخاب کرده باشید تمام شرایطی که در ستون Criteria بنویسید به عنوان شرایط Having در نظر گرفته می شوند. برای اینکه این محدودیتها به شرط Where اضافه گردند، بر روی پانل مشبک متناظر کلیک کرده و از لیست مورد نظر گزینه where را بجای Group by انتخاب کنید.

تذکر: در ستون Alias شما می توانید یک اسم مستعار برای این ستون در نظر بگیرید.

برای دیدن نتایج دید بر روی علامت (!) کلیک نمایید.

توجه داشته باشید که ارتباطها در صورتی برقرار می شود که کلیدهای خارجی بر روی جداول وجود داشته باشد. گرچه شما می توانید با انتخاب فیلد مورد نظر و حرکت ماوس در حالیکه کلیک سمت چپ را فشار داده اید مابین دو جدول ارتباط را برقرار کنید.

۱-۴- آجاکس^۱:

۱-۴-۱- مقدمه ای بر آجاکس:

امروزه معمولاً وقتی صحبت از طراحی برنامه های تعاملی می شود، بیشتر توجه ها معطوف به وب است. برنامه های کاربردی تحت وب به سرعت در حال رشد و توسعه هستند و کاربران مایل هستند که به همان سرعت که با برنامه های کاربردی روزمره^۲ کار می کنند با برنامه های کاربردی وب^۳ هم کار کنند. طراحان برنامه های کاربردی وب هم برای حل این مشکل، روش های طراحی گوناگونی را پیشنهاد کرده اند. یکی از جدیدترین روش ها، روش آجاکس است.

لازم است در ابتدای آشنایی با این روش ابتدا نگاهی داشته باشیم به Suggest Google و Google Maps. در Google Maps وقتی روی قسمتی از نقشه بزرگنمایی^۴ می کنید و یا بالا و پایین می روید همه چیز تقریباً به طور همزمان و بدون بارگذاری مجدد صفحه انجام می شود. در Suggest Google نیز وقتی مشغول تایپ کلمه مورد نظر خود هستید، به طور همزمان کلمات مرتبط با آن و نیز نتایج حاصل از جستجوی آن ها نمایش داده می شود.

¹ Ajax

² Desktop Application

³ Web Application

⁴ Zoom

با توجه به مثال‌های بالا در می‌یابیم که آجاکس فناوری است که در آن با بار شدن یک صفحه، امکان فراخوانی برنامه‌های سمت سرور بدون بارگزاری مجدد صفحه وجود دارد و همین امر باعث بالا رفتن سرعت برنامه‌های تحت وب مبتنی بر آجاکس شده‌است.

۱-۴-۲- آجاکس چیست؟

واژه Ajax مخفف Asynchronous JavaScript And Xml و به مفهوم استفاده نامتقارن JavaScript و Xml بوده است. جالب است بدانید که مقدمات استفاده از این تکنیک از سال‌ها پیش در مرورگرهای وب قرار داده شده بود ولی با وجود استفاده مایکروسافت بصورت محدود در Outlook کسی به موارد استفاده آن بصورت گسترده پی نبرده بود. در حقیقت تاریخچه واقعی آجاکس به عنصری مخفی به نام IFrame در Html بر می‌گردد که با استفاده از آن، برنامه نویسان قادر به انجام اعمالی شبیه آجاکس بودند.

جهت درک بهتر مفهوم آجاکس مثال‌های ملموس زیادی وجود دارد. به عنوان مثال فرض کنید که در جعبه پیام‌های دریافتی^۱ ایمیل خود در Gmail قرار گرفته اید، به محض دریافت یک ایمیل جدید بدون اینکه شما بر روی دکمه و یا لینکی کلیک کرده باشید، خبر دریافت ایمیل جدید در همان صفحه وب به شما داده می‌شود. توجه داشته باشید که تا پیش از ظهور تکنیک آجاکس، جهت مطلع شدن از تغییرات در صفحه وب، حتما می‌بایست صفحه بارگزاری مجدد شده و یا بر روی لینکی کلیک می‌نمودید.

با استفاده از این تکنیک کد نویسی در وب، دیگر لازم به بارگزاری مجدد کل صفحه نیست، بلکه با روش‌هایی می‌توان تنها بخشی از یک صفحه وب را به روز رسانی کرده و بخش‌های دیگر را به قوه خود باقی گذاشت.

برای درک بهتر موضوع به مثالی دیگر می‌پردازیم. ممکن است بارها از دیکشنری‌های آنلاین^۲ چند زبانه در اینترنت استفاده کرده و با آنها آشنایی کافی داشته باشید. لغت مورد نظر خود را در یک جعبه متن^۳ وارد کرده و دکمه جستجو را می‌فشارید. سپس معنای لغت مورد نظر با بارگزاری مجدد صفحه نمایان می‌گردد. متأسفانه در این مثال نیز از آجاکس استفاده نشده اما اگر با بهینه سازی وب سایت، از تکنیک آجاکس استفاده گردد، به راحتی و با تایپ کردن همزمان شما در جعبه متن مورد نظر، لیستی پویا از

^۱ Inbox

^۲ Online

^۳ Textbox

لغات یافت شده (با حذف یا اضافه کردن یک حرف در جعبه متن جستجو، محتوای لیست نیز تغییر می‌یابد)، بدون بارگزاری مجدد صفحه نمایان می‌گردد.

ذکر این نکته خالی از لطف نیست که بدانیم شرکت گوگل^۱ یکی از پیشگامان اصلی استفاده از تکنیک آجاکس در تولیداتی همچون Gmail، Google Suggestion، Google Maps، Google Earth و غیره می‌باشد.

نکته مهم:

شناخت اینکه یک صفحه وب با تکنیک آجاکس و یا بدون آن تولید شده، برای یک برنامه نویس آشنا به جاوااسکریپت^۲ و مباحث کدهای سمت سرور/مشرتی^۳ شاید چندان مشکل نباشد ولی قطعاً یک برنامه نویس نوپا و یا کاربر عادی، در بعضی مواقع قادر به شناسایی آن نخواهد بود.

به عنوان مثال یک صفحه پویا^۴ را در مقابل چشمان خود تجسم کنید. درون این صفحه، چند منو بصورت لینک قرار گرفته که با قرار گرفتن ماوس بر روی هر یک از این لینک‌ها، یک متن مجزا در سمت چپ صفحه ظاهر شده و با کنار گرفتن ماوس از روی لینک، متن مورد نظر نیز محو می‌شود (این عمل بدون بارگزاری مجدد صفحه انجام می‌شود) ولی باید توجه داشت که به هیچ عنوان کد آجاکس در این صفحه مورد استفاده قرار نگرفته، بلکه متن‌های مورد نظر که همیشه یکسان و ثابت هستند، با استفاده از جاوااسکریپت (به روش سمت مشرتی^۵)، بواسطه لایه‌ها و یا هر عنصر دیگری، آشکار و پنهان می‌گردند.

نکته مهم اینکه هرگاه یک صفحه وب عادی بدون بارگزاری مجدد، اعمالی که مرتبط با کدهای سمت سرور^۶ (مانند ارتباط با بانک اطلاعاتی در سرور) است را انجام دهند، به عنوان صفحات مزین به آجاکس محسوب می‌گردند.

مفهوم عملکرد آجاکس:

در این بخش تصمیم گرفتیم تا شما را با مفاهیم فنی و روش برنامه‌نویسی آجاکس آشنا کنیم. برای نوشتن ساده‌ترین کد آجاکس می‌بایست به مباحث HTML, JavaScript, Dom و یک زبان سمت سرور مانند ASP, PHP, ASP.NET و

¹ Google

² JavaScript

³ Client/Server Side

⁴ DHTML

⁵ Client Side

⁶ Server-Side

غیره آشنایی داشته باشید، اگرچه جهت گسترش یک برنامه آجاکس پیچیده، داشتن دانش کافی در رابطه با XML و CSS نیز از ملزومات است.

همانطور که تقریباً همه برنامه نویسان وب مطلع هستند، هر مرورگر وب مانند IE, Firefox, Opera, Mozilla, Netscape و غیره دارای تعداد زیادی شیء درونی به صورت API در زمینه‌های مختلفی چون DOM, CSS, HTML, XML و غیره می‌باشند. (جهت مشاهده لیستی از این اشیا، آدرس زیر را ملاحظه فرمایید)

<http://www.xulplanet.com/references/objref/>

این اشیا به طور ذاتی در مرورگرها وجود داشته و نیازی به تولیدشان نیست، بلکه جهت استفاده از آنها ابتدا می‌بایست عملکرد آن شیء خاص را مورد بررسی قرار داده و سپس توسط یک زبان سمت مشتری مانند جاوااسکریپت، به راحتی از آن استفاده نمود.

نکته مهم:

نکته قابل توجه در این رابطه، عدم هماهنگی و استاندارد نبودن نام این اشیا در مرورگرهای مختلف می‌باشد که البته یکی از دغدغه‌های آجاکس نیز همین موضوع می‌باشد. این مسئله نه تنها در مورد اشیا، بلکه در مورد تگ‌های HTML نیز صادق است. به عنوان مثال استفاده یک تگ جدید و ناشناخته در مرورگر X دلیلی بر کار کردن همان تگ در مرورگر Y نخواهد بود.

شاید جالب باشد اگر بدانید، سناریوی اصلی آجاکس حول محور یک شیء به نام XMLHttpRequest می‌چرخد که آن را با نام مخفف XHR می‌شناسند. این شیء بازیگر نقش اول و تقریباً همه کاره کدهای آجاکس بوده و البته طبق گفته قبلی مبنی بر استاندارد نبودن مرورگرهای وب، با نام‌های متفاوت ولی عملکرد یکسانی، قابل دسترس و استفاده است. نحوه عملکرد این شیء و متدها و خصوصیات آن، در ادامه مورد بررسی قرار خواهد گرفت.

نکته قابل توجه این است که این شیء در یک کد سمت مشتری و با استفاده از جاوااسکریپت اجرا شده و سپس در زمان معین، یک کد سمت سرور را بدون متوجه شدن کاربر سایت، در سرور اجرا کرده و نتیجه اجرای کد مربوطه را از سرور گرفته و مجدداً بدون بارگزاری مجدد صفحه در اختیار مرورگر قرار می‌دهد. سپس جاوااسکریپت نتیجه بازگشتی را به دلخواه برنامه نویس، مورد استفاده قرار داده و یا در قسمتی از صفحه مقابل کاربر بدون بارگزاری مجدد نمایان می‌سازد.

۱-۴-۳- کدنویسی آجاکس:

برای کدنویسی آجاکس، ابزارهای آماده بسیاری در محیط‌های نرم افزاری IDE (مانند visual Studio.NET) فراهم گردیده‌است. در ادامه به نحوه استفاده از این ابزارها خواهیم پرداخت. علاوه بر ابزارهای گفته شده می‌توانیم برای پیاده سازی ساده آجاکس از ساده‌ترین ابزار مانند یک ویرایشگر متن مانند Notepad استفاده کرد. در اینجا ساده‌ترین کد آجاکس را با توضیحات کامل ارائه خواهیم کرد.

مرحله اول:

ابتدا در یک ویرایشگر صفحات وب مانند Expression Web و یا یک ویرایشگر متن ساده یک فایل جدید به نام Ajax.htm ایجاد نموده و کدهای جاوااسکریپت و HTML زیر را در آن قرار دهید.

```
<Html>
<Body>
<script type="text/JavaScript">
Function ajaxFunction ()
{
Var xmlhttp=new ActiveXObject ("Msxml2.XMLHTTP")?
New ActiveXObject ("Microsoft.XNLHTTP"): new XMLHttpRequest ();
XmlHttp. Onreadystatechange=function () {
If (XmlHttp.readyState==4){
document.AjaxForm.Time.value=XmlHttp.responseText;
}
}
xmlHttp.open ("Get","time.asp", true);
xmlHttp.Send (null);
}
<form name="AjaxForm">
Name :< input type="text" onchange="ajaxFunction () ;">
Time :< input type="text" name="time">
</form>
</body></html>
```

¹ Toolkit

توضیح کدهای فوق:

در ابتدا برای قرار دادن یک کد جاوااسکریپت در یک صفحه وب، می بایست از تگ‌های `<script>` و `</script>` استفاده نمود و کدهای جاوا را در میان این دو تگ قرار دهیم تا مرورگر به هنگام اجرای این بخش از صفحه وب، از مفسر جاوا برای اجرای آن استفاده نماید. در ضمن فراموش نکنید که کدهای جاوااسکریپت حساس به متن بوده و به حروف کوچک و بزرگ حساس هستند.

سپس یک تابع به نام `ajaxFunction` تعریف می‌گردد که عملی را انجام داده و هیچ خروجی خاصی به فراخواننده خود ندارد. در نظر داشته باشید که کد فراخوان این تابع در ادامه فایل `HTML` قرار دارد. همانطور که قبلاً ذکر شد شیء `XMLHttpRequest` وجود دارد که تمامی عملیات آجاکس را به عهده خواهد گرفت. شیء `XMLHttpRequest` دارای متدها و خصوصیت‌هایی می‌باشد که در انتهای این بخش به اختصار توضیح داده خواهد شد. یکی از خصوصیات این شیء، خصوصیتی به نام `Onreadystatechange` است، که به عنوان یک رویدادگرا^۱ مورد استفاده قرار می‌گیرد. این رویداد، پس از فراخوانی یک کد طرف سرور، بلافاصله به طور خودکار فراخوانی شده و کدهای داخل آن اجرا می‌شود.

یکی دیگر از خصوصیات این شیء، خصوصیتی به نام `Ready State` است که از نوع عددی بوده و نشان دهنده عملکرد سرور در ارسال پاسخ می‌باشد. در این خصوصیت عددی ما بین ۰ تا ۴ (بصورت خودکار) قرار خواهد گرفت که هر عدد مطابق با جدول زیر معنا و مفهوم خاص خود را دارد.

جدول شماره ۱-۱- مقادیر خصوصیت `Ready state`

۰	درخواستی که به سمت سرور ارسال شده، هنوز کار خود را آغاز نکرده است.
۱	درخواست در سرور در حال بررسی و اجرا است.
۲	درخواست از سرور برای <code>Client</code> ارسال شده است.
۳	درخواست در حال ارسال توسط <code>Client</code> است.
۴	عمل پاسخ‌گویی به درخواست اتمام یافته است.

همانطور که ملاحظه می‌کنید، با استفاده از این خصوصیت به راحتی وضعیت درخواست ارسالی به سرور، قابل پیگیری است.

^۱ Event Handler

یکی دیگر از خصوصیات این شیء خصوصیتی به نام Response Text است که از نوع رشته می باشد که نتیجه ارسالی از سمت سرور در آن قرار می گیرد.

هم اکنون به بررسی دو متد بسیار مهم به نام های open, send که عملیات ارسال درخواست به سمت سرور را انجام می دهند، می پردازیم.

```
xmlHttp.open ("GET", "time.asp", true);
```

```
xmlHttp.send (null);
```

متد open وظیفه اعمال تنظیمات قبل از ارسال و متد send وظیفه ارسال درخواست را بر عهده دارد و با توجه به اینکه این دو خط کد، بعد از بلوک رویداد onreadystatechange قرار گرفته، ولی تا زمانی که متد Send شیء XMLHttpRequest فراخوانی نگردیده، Event Handler فراخوانی نخواهد شد.

متد open دارای دو پارامتر ضروری (دو پارامتر اول) و یک پارامتر غیر ضروری (آخرین پارامتر) می باشد. در اولین پارامتر این متد می توان از دو مقدار GET, POST استفاده نمود که البته هر یک از این مقادیر در مواقع خاص خود کاربرد دارند. دومین مقدار نیز نام و مسیر یک فایل طرف سرور را مانند ASP, PHP, ASPX, CFM و غیره را تعیین می نماید که در مثال فوق، نام فایلی به نام time.asp در همان پوشه فایل HTML، تنظیم شده است.

سومین پارامتر (که به صورت پیش فرض مقدار آن true می باشد)، وضعیت نامتقارن بودن و یا نبودن Event Handler مربوطه را مشخص می کند. در صورت قرار دادن مقدار True پس از اجرای متد send بلافاصله رویداد onreadystatechange نیز فراخوانی می گردد، اما در صورت قرار دادن مقدار False، ابتدا منتظر اتمام پردازش سرور و ارسال پاسخ شده و سپس رویداد مورد نظر فراخوانی می گردد. در نظر داشته باشید که در صورت استفاده از مقدار false، اگر به یکباره سرور دچار مشکلی گردیده و یا دریافت پاسخ از سرور تعویق افتد، نرم افزار وب مدت کوتاهی مختل شده و کنترل از دست برنامه نویس خارج می شود. البته متد open دارای دو پارامتر غیر ضروری دیگر است که در این بخش به آن نمی پردازیم. و بالاخره متد send، مقداری را به صورت پارامتر به سمت سرور ارسال می کند که در مثال فوق به دلیل عدم نیاز به ارسال مقادیری به سرور از واژه null استفاده گردیده است.

حال به توضیح فرم وب در کد HTML خواهیم پرداخت.

```
<form name="AjaxForm">
```

```
Name :< input type="text"onchange="ajaxFunction ();">
```



```
Time :< input type="text" name="time">
```

```
</form>
```

همانطور که در کد HTML فوق ملاحظه می فرمایید، فرمی به نام AjaxForm با ۲ عدد جعبه متن^۱ که نام یکی از آنها time می باشد، تعریف شده است. در رویداد onchange اولین جعبه متن، تابع ajaxFunction فراخوانی می شود.

هم اکنون یک فایل جدید به نام time.asp در ویرایشگر متن ویا وب ایجاد کرده و کدهای زیر را عینا در آن وارد کنید.

```
<%
```

```
Response.expires=-1
```

```
Response.Write (time)
```

```
%>
```

کد بسیار ساده فوق، ساعت سرور را از طریق تابع داخلی Time (یکی از توابع VBscript) گرفته و بواسطه متد Write در

خروجی نمایش می دهد. هم اکنون فایل Ajax.htm, time.asp را در یک پوشه سرور وب قرار داده و فایل Ajax.htm را اجرا نمایید.

همانطور که قبلا اشاره شد، شیء XMLHttpRequest دارای متدها و خصوصیات دیگری نیز می باشد که برخی از مهمترین آنها را به همراه توضیح مختصری در جدول زیر ملاحظه می فرمایید.

۱-۴-۴- مزایا و معایب آجاکس:

هر عنصر جاندار و بی جانی در این کره خاکی دارای مزایا و معایبی می باشد. آجاکس نیز به نوبه خود ترازوی مقایسه ای دارد

که تقریبا دو کفه مزایا و معایب آن با یکدیگر برابری می کند. در این بخش به این موارد بطور تقریبا کلی اشاره خواهیم کرد.

الف) مزایا:

۱- تولید نرم افزارهای تحت وب با خصلت Desktop:

ایده پیاده سازی نرم افزارهای Desktop در محیط وب، امروزه در حال گسترش است و تکنیک آجاکس یکی از مناسب ترین روش های معقول پیاده سازی این ایده می باشد. اگرچه فایل های فلش نیز امکاناتی مانند آجاکس در اختیار برنامه نویسان قرار می دهند ولی همانند آجاکس چندان مورد توجه قرار نگرفته اند.

¹ Textbox

جدول شماره ۱-۲- متدها و خصوصیات شیء XMLHttpRequest

<p>با استفاده از این متد، تمامی عملیات درخواست از سرور در هر مرحله‌ای، بلافاصله Cancel خواهد شد.</p>	<p>Abort()</p>	<p>متد^۱</p>
<p>خصوصیتی از نوع عددی است که وضعیت درخواست HTTP به سرور را نمایش می‌دهد. به عنوان مثال قرار گرفتن عدد ۴۰۴ در این خصوصیت نشان‌دهنده عدم وجود آدرس فایل در متد open می‌باشد. اما عدد ۲۰۰ دلیلی بر وجود این فایل در سرور است.</p>	<p>status</p>	<p>خصوصیات</p>
<p>این خصوصیت از نوع رشته ای است که وضعیت درخواست HTTP به سرور در قالب یک متن در آن مقداردهی می‌گردد.</p>	<p>status Text</p>	
<p>همانند response Text می‌باشد، با ایت تفاوت که مقدار خروجی در قالب XML به کلاینت باز می‌گردد که به صورت DOM می‌توان به تگ‌های آن دسترسی داشته و بخشی از محتویات XML را مورد استفاده قرار داد.</p>	<p>Response XML</p>	

۲- استفاده کمتری از پهنای باند سرور:

پهنای باند یک سرور برای وب سایت‌هایی که بازدیدکنندگان بسیاری دارند، از اهمیت بالایی برخوردار است. به عنوان مثال سایت سنجش آموزش کشور (www.sanjesh.org) را در اولین روز ارائه نتیجه کنکور، جهت سرویس دهی به صدها هزار بازدید کننده مجسم کنید. هر کاربر با وارد شدن به سایت و مراجعه به بخش‌های مربوطه، باعث ایجاد درخواست‌هایی برای وب سرور شده و با اشغال پهنای باند، باعث افت سرعت خواهد شد. معمولاً مدیران سایت‌ها در چنین مواقعی با ارتقاء پهنای باند سرور یا ارتقای سخت افزاری، سعی در بهینه سازی سرویس دهی دارند. اما نکته قابل توجه این است که با استفاده از تکنیک آجاکس،

¹ Method

میزان درخواست‌های ارسال شده به سرور و حجم پاسخ‌های ارسال شده از سرویس دهنده، به شدت کاهش یافته و شاید دیگر نیازی به ارتقا نباشد.

۳- سرعت بالای برنامه‌های مبتنی بر آجاکس:

سالهست که انتظار جهت بارگزاری شدن یک صفحه و قطع ارتباط با سرور، پس از کلیک شدن بر روی یک لینک، کاربران را با دغدغه‌هایی مواجه کرده‌است. با استفاده از تکنیک آجاکس علاوه بر سرعت بخشیدن به عملکرد کاربر، باعث زیبایی کارکرد نرم افزار و رضایتمندی کاربران نیز خواهد شد.

(ب) معایب:

۱- پیچیدگی کدنویسی:

استفاده از تکنیک آجاکس ظاهرا یک تکنیک دوست داشتنی است ولی در عمل کدنویسی پیشرفته جهت انجام اعمالی در صفحه وب، باعث ایجاد کدهای پیچیده می‌گردد. در بسیاری مواقع که تقریبا بیش از ۸۰ درصد یک صفحه وب پس از بارگزاری مجدد می‌بایست تغییراتی داشته باشند، از تکنیک آجاکس استفاده نمی‌گردد. بنابراین تیم طراح و برنامه نویس با در نظر گرفتن نقاط قوت و ضعف و وضعیت موجود، تصمیم به استفاده و یا عدم استفاده از آجاکس می‌گیرند. لازم به ذکر است که زیاده روی در استفاده از آجاکس بعضا باعث افت کارایی نرم افزار وب می‌گردد.

۲- عدم سازگاری مرورگرها:

همانطور که پیش از این اشاره شد، به دلیل عدم وجود استاندارد ما بین مرورگرهای وب، پیشنهاد می‌شود که حتما کد آجاکس خود را با آخرین نسخه مرورگرهای رایج دنیا امتحان نموده تا از صحت عملکرد آن‌ها اطمینان حاصل کنید.

۳- عدم کاربرد دکمه‌های Back, Stop, Refresh:

همانطور که می‌دانید، مرورگرها همیشه آدرس صفحه قبلی را در حافظه نگاه داشته و با فشردن دکمه Back به آن صفحه مراجعه می‌کنند. اما یک صفحه وب را در نظر بگیرید که با استفاده از تکنیک آجاکس بدون منتقل شدن به صفحه بعد، عملی را در همان صفحه انجام داده و سپس نتیجه را نمایش خواهد داد. اساسا در این سناریو هیچ صفحه قبلی وجود ندارد که با فشردن

دکمه Back به آن مراجعه نموده و بنابراین اطلاعات صفحه وب پس از تغییر محتوا، غیر قابل بازگشت خواهد بود. این موضوع برای دکمه‌های Stop و Refresh نیز صادق خواهد بود.

۴- اتکای بیش از حد به زبان جاوااسکریپت:

تقریباً همه مرورگرهای وب از زبان جاوااسکریپت به عنوان یک زبان سمت مشتری پشتیبانی می‌کنند. بنابراین این زبان برنامه‌نویسی، نقشی بسیار مهم و کلیدی در استفاده از تکنیک آجاکس بازی می‌کند. در این میان اگر به طور اتفاقی اجرای کدهای جاوااسکریپت غیر فعال شده باشد، عملاً آن صفحه وب قابل استفاده کاربر مورد نظر نخواهد بود.

۱-۴-۵- نصب آجاکس:

برای نصب آجاکس ابتدا باید به سایت <http://www.asp.net/Ajax> مراجعه کرده و نرم افزار AspAjaxExtSetup را دانلود کنید. علاوه بر این نرم افزار شما می‌توانید ابزارهای آماده آجاکس را نیز دانلود کنید. اگر دانلود نرم افزارهای بالا تمام شده‌است، حال می‌توانید مراحل نصب آجاکس و همچنین نصب ابزارهای آجاکس را شروع کنیم.

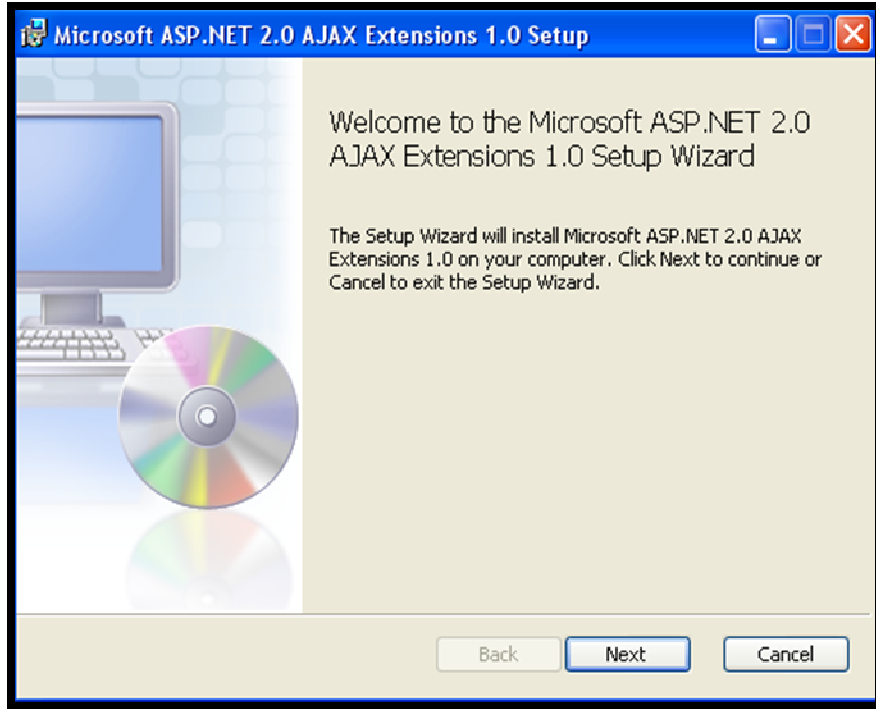
مرحله ۱: بر روی AspAjaxExtSetup کلیک می‌کنیم، پنجره ای مانند شکل ۱-۱ باز می‌شود. این صفحه خوش آمد گویی آجاکس است. سپس بر روی next کلیک کرده و وارد مرحله بعدی می‌شویم.

مرحله ۲: صفحه بعدی که پیش روی شما است صفحه مجوز^۱ است. شما با تیک زدن checkbox پائین صفحه موافقت خود را اعلام می‌کنید سپس بر روی next کلیک کنید. (مطابق شکل ۱-۲)

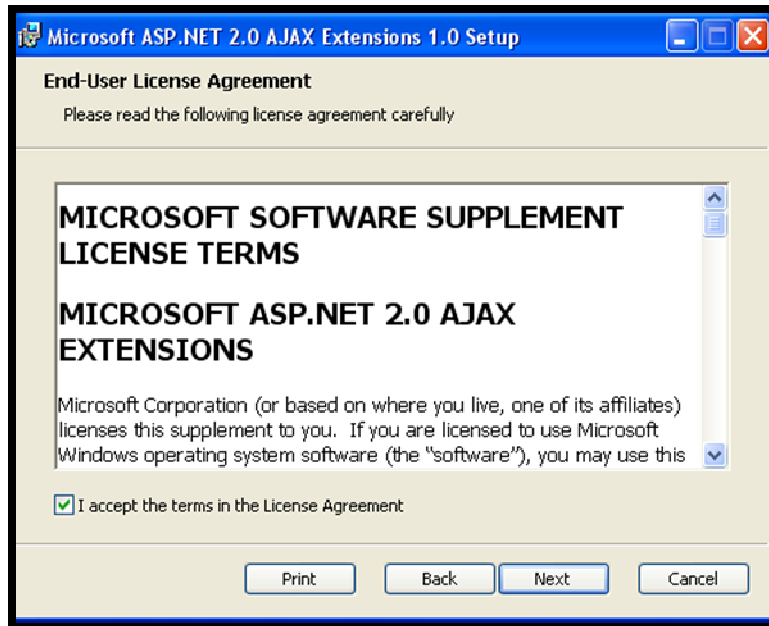
مرحله ۳: در این صفحه بر روی Install کلیک کنید تا نصب آجاکس بر روی سیستم شما آغاز شود. (مطابق شکل ۱-۳)

مرحله ۴: پس از اینکه نصب آجاکس بر روی سیستم شما کامل شد، پنجره ای مانند شکل شماره ۴ باز می‌شود. شما می‌بایست بر روی گزینه Finish کلیک کنید.

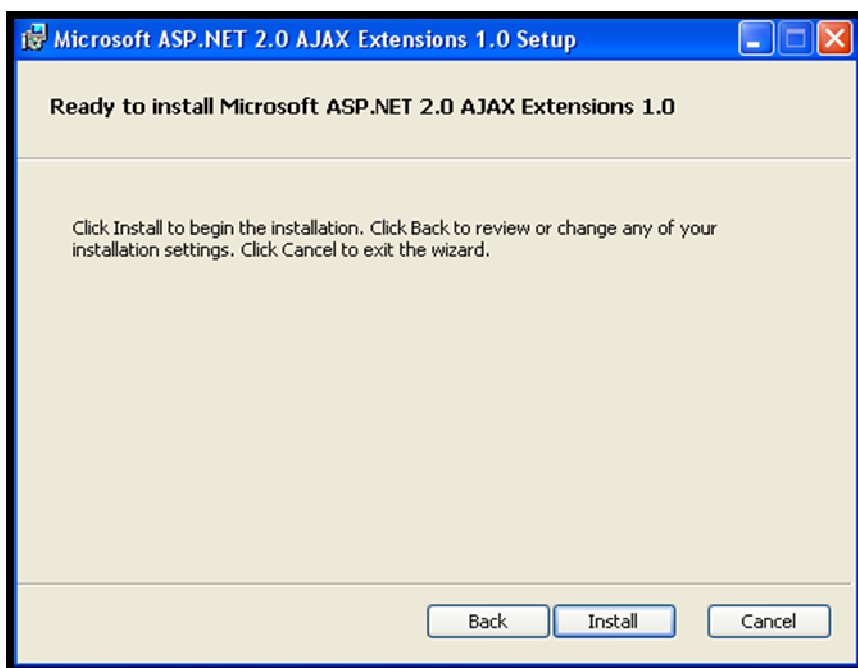
¹ License



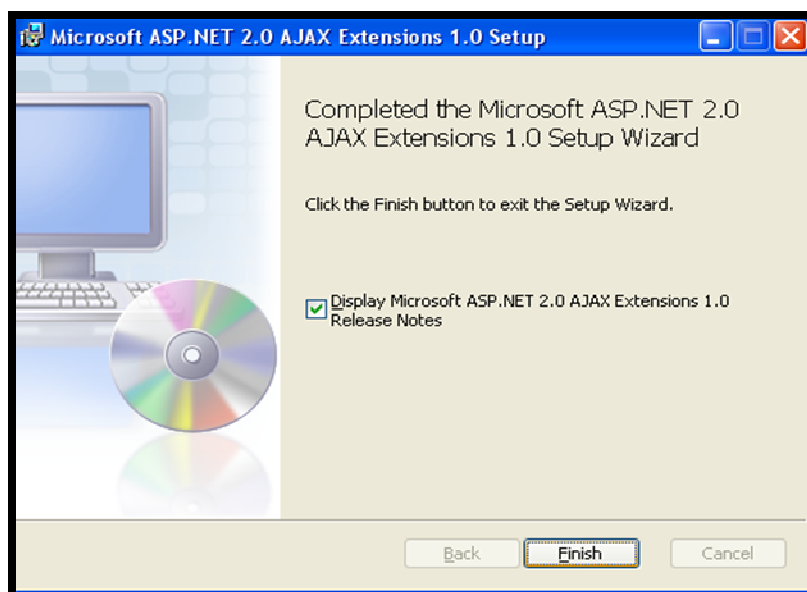
شکل ۱-۱- اولین مرحله نصب آجاکس



شکل ۱-۲- دومین مرحله نصب آجاکس



شکل ۱-۳- سومین مرحله نصب آجاکس

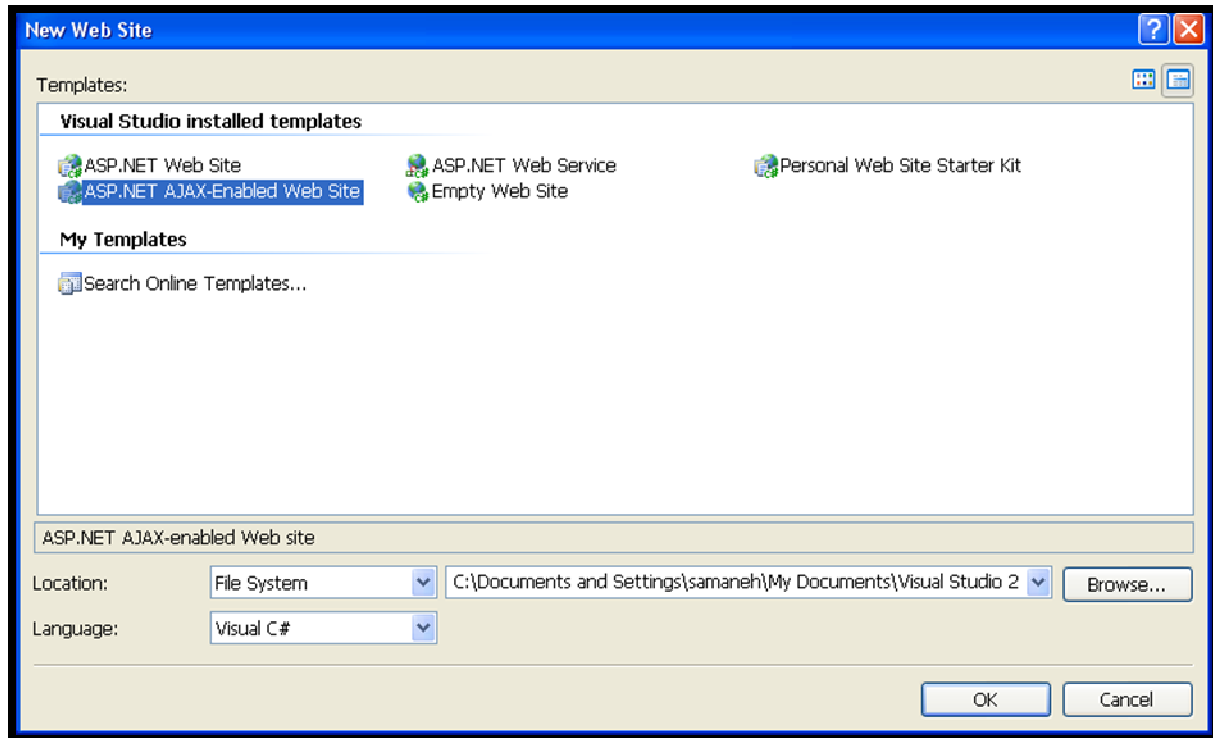


شکل ۱-۴- چهارمین مرحله نصب آجاکس

– برای برداشتن^۱ آجاکس می توانیم مانند دیگر نرم افزارهای دیگر وارد Control Panel شویم، سپس بر روی Add/Remove Program کلیک کنید، سپس آجاکس را عزل کنید.

¹Remove

پس از نصب آجاکس شما می توانید صفحاتی را تولید کنید که مبتنی بر آجاکس هستند. برای این منظور نرم افزار Visual Studio را باز نموده از منوی File گزینه new website را باز نموده سپس مطابق شکل شماره ۱-۵ در قسمت Templates گزینه Asp.net Ajax_Enabled Website را انتخاب کنید. حال بر روی OK کلیک کنید.



شکل ۱-۵- تولید صفحه وب مبتنی بر آجاکس

صفحه سفیدی ایجاد خواهد شد که حاوی ابزاری به نام Script Manager است. این ابزار برای کنترل تمامی کدهای آجاکس و یا ابزارهای آماده آجاکس لازم و ضروری است.

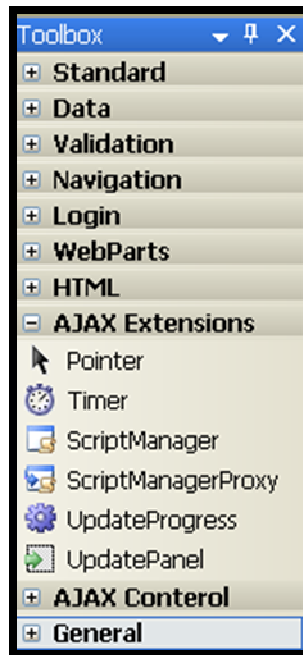
برای مطمئن شدن از نصب کامل آجاکس می توانید از روش زیر استفاده کنید. نرم افزار Visual Studio را باز کنید از منوی View گزینه Toolbox را انتخاب کنید. با انتخاب این گزینه پنجره‌ای حاوی تمام ابزارهای این نرم افزار نمایش داده خواهد شد. حال در این پنجره منویی با نام Ajax Extensions را جستجو کنید، اگر یافت شد آجاکس به طور کامل نصب شده است.

منوی Ajax Extension با نصب آجاکس به صورت خودکار در جعبه ابزار^۱ نرم افزار Visual Studio قرار می گیرد که این منو دارای ۶ عدد ابزار با نام‌های زیر است:

- Pointer

¹ Toolbox

- Timer
- Script Manager
- Script Manager Proxy
- Update Progress
- Update Panel



شکل ۱-۶- ابزارهای اصلی آجاکس

۱-۴-۶- نصب ابزارهای آجاکس:

همانطور که قبلا اشاره شده آجاکس دارای ابزارهای آماده ای است که هر یک از آنها دارای ویژگی های خاصی هستند و می توانند به شما در طراحی سایت کمک کنند. برای استفاده از این ابزارها شما ابتدا آجاکس را نصب نموده سپس با طی مراحل زیر این ابزارها را به Visual Studio اضافه می نمایید.

مرحله ۱: به آدرس زیر مراجعه کنید و فایل Ajax Control Toolkit را دانلود کنید. در هنگام دانلود کردن شما می بایست آدرس مکانی که می خواهید فایل در آنجا ذخیره شود را C:\Program Files\Microsoft ASP.NET قرار دهید.

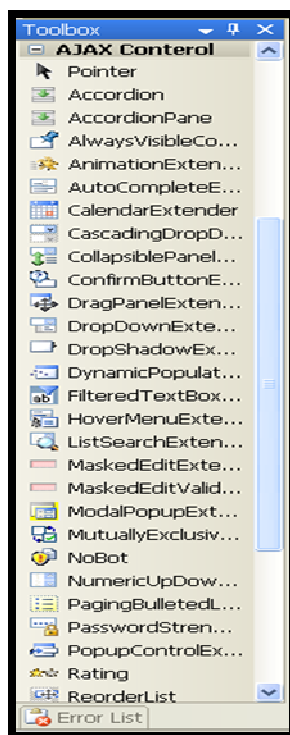
مرحله ۲: پس از دانلود کردن، نرم افزار Visual Studio را باز کرده و از منوی View گزینه toolbox را انتخاب کنید. سپس در پنجره جعبه ابزار بر روی یکی از تب‌ها^۱ کلیک راست کرده و از منویی که باز می‌شود گزینه Add Tab را انتخاب کنید. تب جدیدی ایجاد می‌شود که شما می‌بایست نام آن را Ajax Control قرار دهید.

مرحله ۳: پس از ایجاد تب جدید بر روی آن کلیک راست کرده و گزینه Choose Item را انتخاب کنید. با انتخاب این گزینه پنجره‌ای با نام Choose Toolbox Items باز می‌شود. در این پنجره بر روی NET Framework Components کلیک کنید. سپس بر روی Browse کلیک کرده و به آدرس زیر رفته و فایل Ajax Control Toolkit.dll را انتخاب کنید سپس open را کلیک کنید.

C:\Program Files\Microsoft ASP.NET\AjaxControlToolkit\Sample Website\Bin\Ajax
Control Toolkit.dll

سپس بر روی ok کلیک کنید.

مرحله ۴: حال نگاهی به جعبه ابزار خود بیندازید مشاهده خواهید کرد که در تب با نام Ajax Control حدود ۳۸ عدد ابزار آجاکس اضافه شده است که شما می‌توانید از آن‌ها استفاده نموده و از آن لذت ببرید. (به شکل شماره ۱-۷ توجه فرمائید)



شکل ۱-۷ ابزارهای کنترلی آجاکس

¹ Tab

نکته مهم:

اگر مایل هستید که به صورت ملموس تری با ابزارهای آجاکس آشنا شوید می توانید مراحل زیر را ببینید:

۱- نرم افزار Visual Studio را باز کنید.

۱- از منوی File گزینه open website را انتخاب کنید.

۲- به آدرس زیر رفته و فایل sample website را انتخاب کنید.

C:\Program Files\Microsoft ASP.NET \AjaxControlToolkit

۳- در قسمت solution explorer بر روی Default.aspx کلیک راست کرده و گزینه View in browser را انتخاب کنید.

پس از انتخاب این گزینه صفحه وب باز می شود که شامل مثال هایی برای ابزارهای آجاکس است. علاوه بر مثال شما می توانید توضیحات کامل هر ابزار به همراه خصوصیت های هر ابزار را مطالعه بفرمائید.

۱-۴-۷- معرفی برخی از ابزارهای آماده آجاکس:

همانطور که قبلا اشاره کردیم آجاکس دارای ابزارهای آماده ای است که به راحتی می توانید از آن ها در ASP.NET

استفاده کنید. در این قسمت سعی می کنیم تا شما را با چند نمونه از این ابزارها و خصوصیات آن ها آشنا کنیم.

۱) ابزار Script Manager:

یکی از مهمترین ابزارهای آجاکس ابزار Script Manager است در واقع این ابزار کنترل کننده دیگر ابزارهای آجاکس

می باشد. در هر صفحه ای که شما از ابزارهای آجاکس استفاده می کنید وجود یک عدد Script Manager ضروری است، و

البته این ابزار را می بایست قبل از دیگر ابزارهای آجاکس قرار دهیم. وظیفه اصلی این ابزار برقراری ارتباط با موتور آجاکس است.

۲) ابزار Update Panel:

وظیفه اصلی این ابزار بروز رسانی تنها بخشی از صفحه است و باعث می شود که فقط پارامترهای مورد نیاز به سمت سرور

فرستاده شوند و از بارگزاری مجدد صفحه جلوگیری می کند.

۳) Drag Panel: این ابزار این امکان را به شما می دهد که پنلی^۱ با قابلیت حرکت کردن^۲ بر روی سایت خود داشته باشید به این صورت که کاربر می تواند این پنل را بنا به سلیقه خود به هر قسمت از صفحه سایت حرکت دهد و مکان جدیدی را برای آن انتخاب کند. این ابزار دارای خصوصیات زیر است که می بایست تنظیم شود.

- **TargetControlID:** در این خصوصیت نام پنلی را که می خواهید قابل حرکت کردن باشد را قرار می دهید.
- **DragHandleID:** در این خصوصیت نام کنترلی را که این پنل برای آن استفاده می شود را قرار می دهید. هنگامی که کاربر بر روی این کنترل کلیک کرده و حرکت دهد حرکت خواهد کرد.

۴) Textboxwatermark: این ابزار به یک عدد جعبه متن متصل خواهد شد و هنگامی که این جعبه متن خالی باشد این ابزار فعال شده و پیغامی را به کاربر نمایش می دهد. محتوای این پیغام و همچنین رنگ متن و فونت متن و رنگ پس زمینه را شما می توانید در یک فایل CSS تنظیم کنید و سپس به این ابزار متصل^۳ کنید. این ابزار دارای خصوصیات زیر می باشد.

- **TargetcontrolID:** نام جعبه متنی را که می خواهید این کنترل برای آن اجرا شود را در این قسمت قرار می دهید.
- **WatermarkText:** متنی را که می خواهید در هنگام خالی بودن جعبه متن نمایش داده می شود در این قسمت وارد نمایید.

- **WatermarkCSSClass:** نام فایلی را که می خواهید این کنترل از آن استفاده کند را وارد نمایید.

لازم به ذکر است که وارد کردن خصوصیات ۱ و ۲ برای این ابزار ضروری است.

۵) ابزار انیمیشن:

این هم ابزاری است جهت اجرای انیمیشن بر روی سایت. این ابزار به یک کنترل خاص ملحق می شود و زمانی که رخدادی مانند `onload` , `onclick` , `onmouseover` و یا `onmouseover` برای این کنترل اتفاق افتد انیمیشن را اجرا می کند.

خصوصیات ابزار انیمیشن:

- **TargetControlID:** نام کنترلی که می خواهید انیمیشن بر روی آن اجرا شود را در این قسمت قرار می دهید.
- **Onload:** ساختن انیمیشن زودتر از اینکه صفحه بارگزاری شود.

¹ Panel
² Drag
³ Attach

- **OnClick**: با کلیک کردن بر روی کنترل، انیمیشن اجرا می شود.
- **OnMouseover**: تولید کردن انیمیشن وقتی ماوس بر روی کنترل عبور می کند.
- **OnMouseover**: تولید کردن انیمیشن وقتی ماوس از کنترل خارج می شود.

Slideshow (۶)

این ابزار به شما این امکان را می دهد که تصاویر خود را به صورت اسلاید بر روی سایت به نمایش گذارید. به علاوه شما می توانید به گونه ای این ابزار را طراحی کنید که بتوانید بر روی تصاویر خود عملیات `play`, `next`, `previous` را انجام دهید.

خصوصیات ابزار:

در زیر به برخی از خصوصیات این ابزار اشاره کرده ایم:

- **Slideshowservicepass**: مسیر وب سرویسی که این ابزار تصاویر را از آنجا می گیرد مشخص می کند.
- **NextButtonID**: نام دکمه ای که به شما اجازه می دهد با کلیک کردن بر روی آن تصویر بعدی را ملاحظه بفرمائید.
- **PlayButtonID**: نام دکمه ای که به شما اجازه می دهد با کلیک کردن بر روی آن اسلایدها را اجرا کنید.
- **PreviousButtonID**: نام دکمه مربوط به اسلاید قبلی را در این قسمت می آوریم.
- **PlayButtontext**: نام دکمه مربوط به اسلاید بعدی را در این قسمت می آوریم.
- **StopButtontext**: متن مخصوص دکمه متوقف شدن اسلایدها را در این قسمت قرار می دهیم.
- **PlayInterval**: زمان توقف بر روی اسلایدها را مشخص می کنیم.
- **ImageTitlelableID**: برای هر یک از تصاویر خود می توانید یک عنوان قرار دهید.
- **ImageDescriptionlableID**: برای هر یک از تصاویر خود می توانید یک متن توضیحی قرار دهید.
- **Loop**: یک متغیر از نوع است بولی^۱ که اگر مقدار آن را برابر با درست^۲ قرار دهید می توانید تصاویر را به صورت چرخشی مشاهده کنید.
- **Autoplay**: یک متغیر از نوع است بولی که اگر مقدار آن را برابر با True باشد به این معنا است که به صورت اتوماتیک اسلایدها اجرا شوند.

¹ Boolean

² True

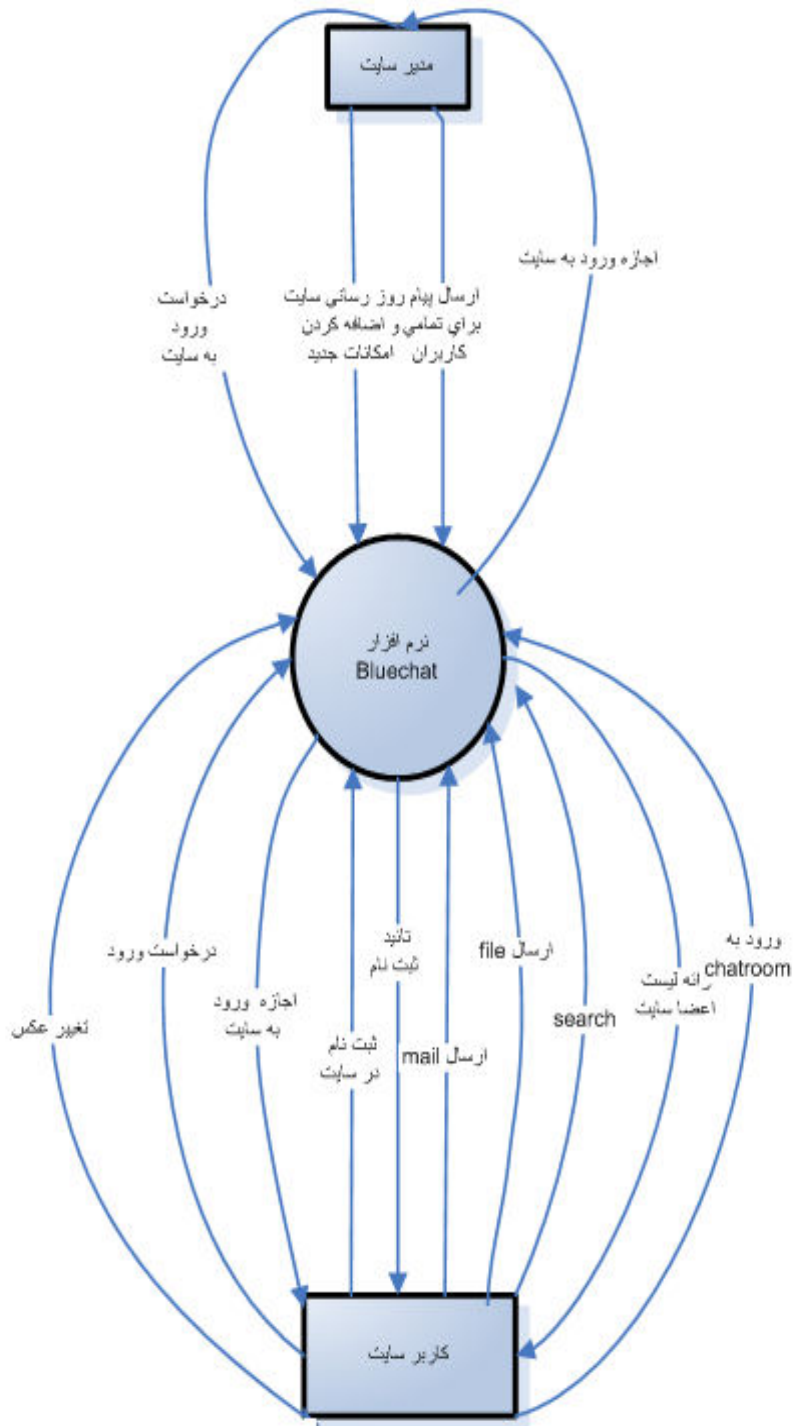
فصل دوم: تجزیه و تحلیل و طراحی سیستم

تحلیل و ترسیم دیاگرام جریان داده‌ها:

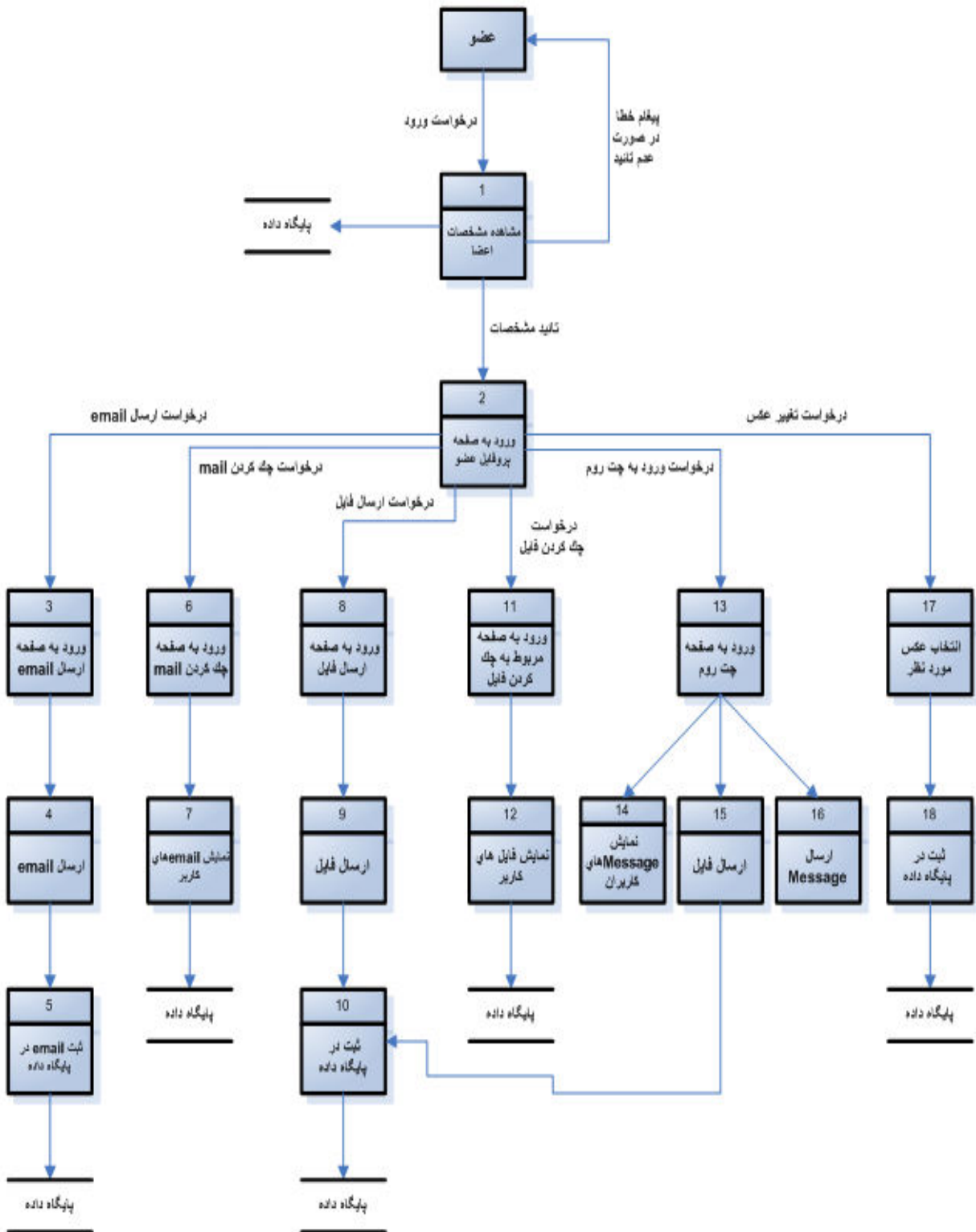
در این بخش به تحلیل و ترسیم تعدادی از دیاگرام‌های جریان داده‌ها می‌پردازیم. و هدف اصلی در این بخش توصیف مراحل

کاری سیستم و روند کاری اعضا در سایت می‌باشد.

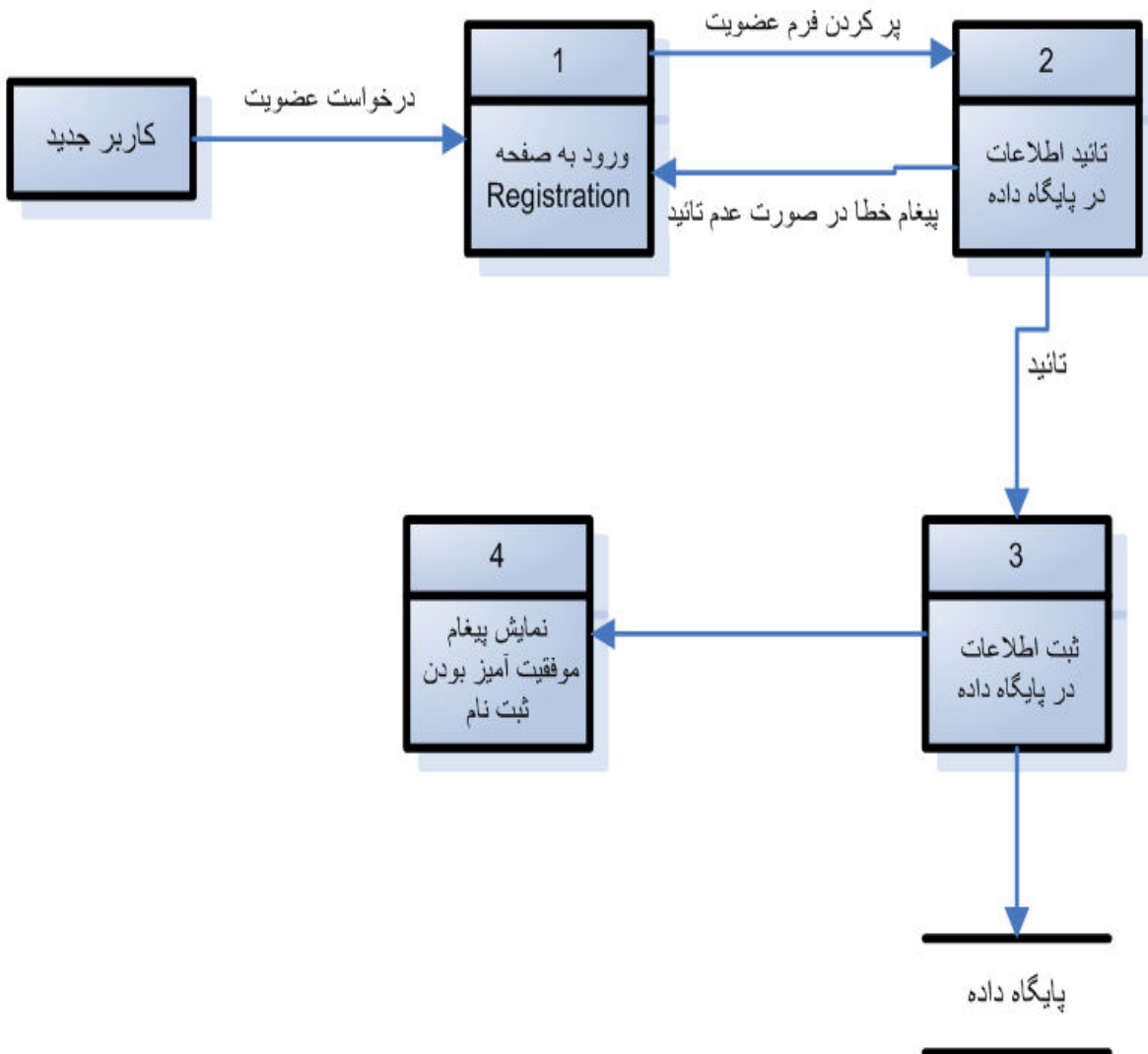
نمودار محتوا:



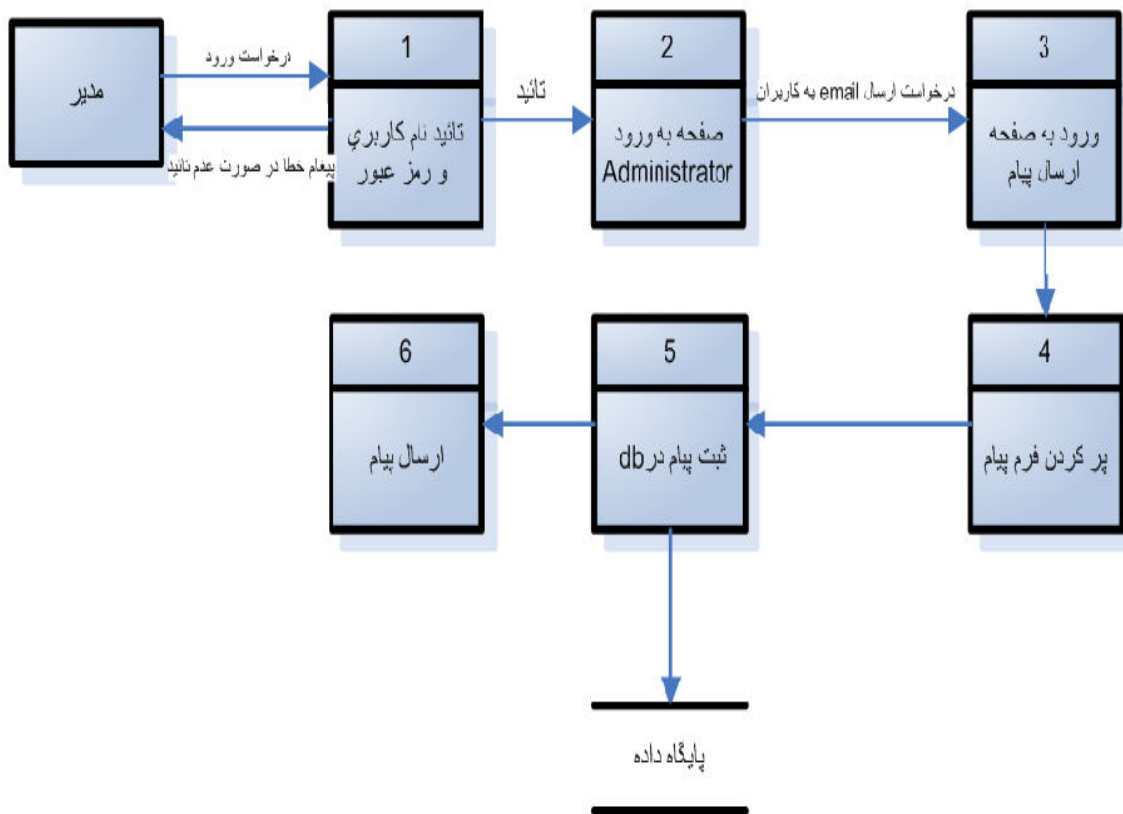
نمودار سطح صفر DFD:



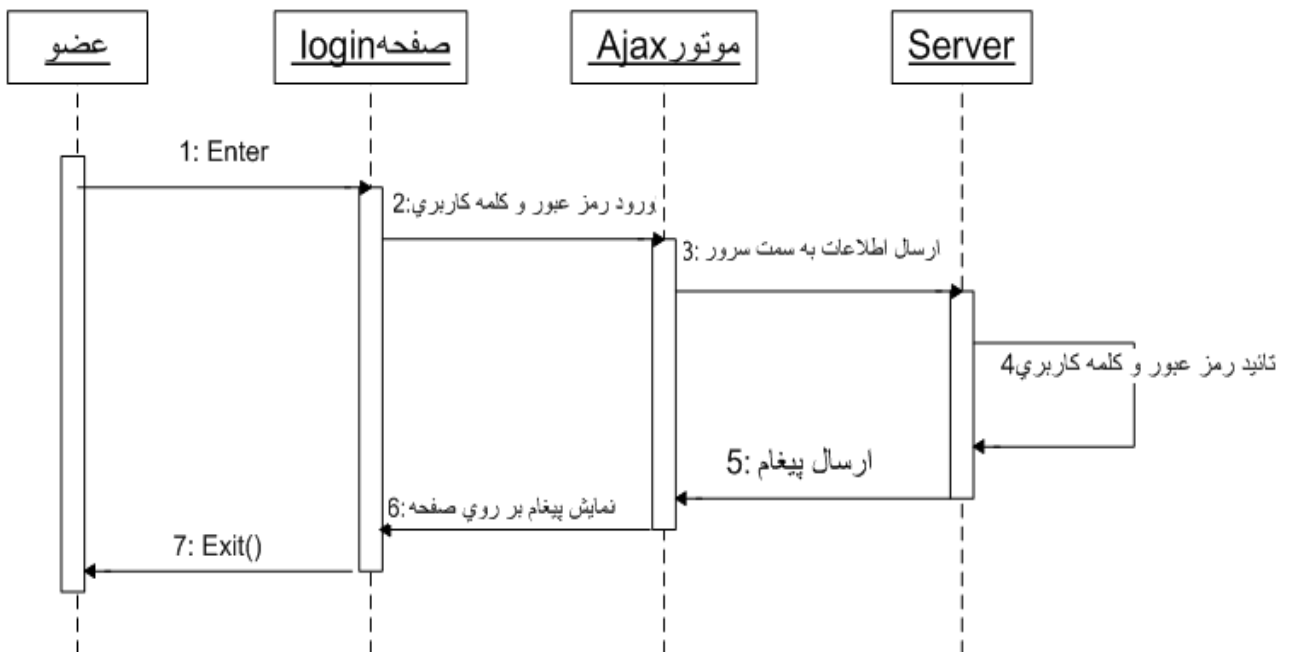
ادامه نمودار سطح صفر DFD :



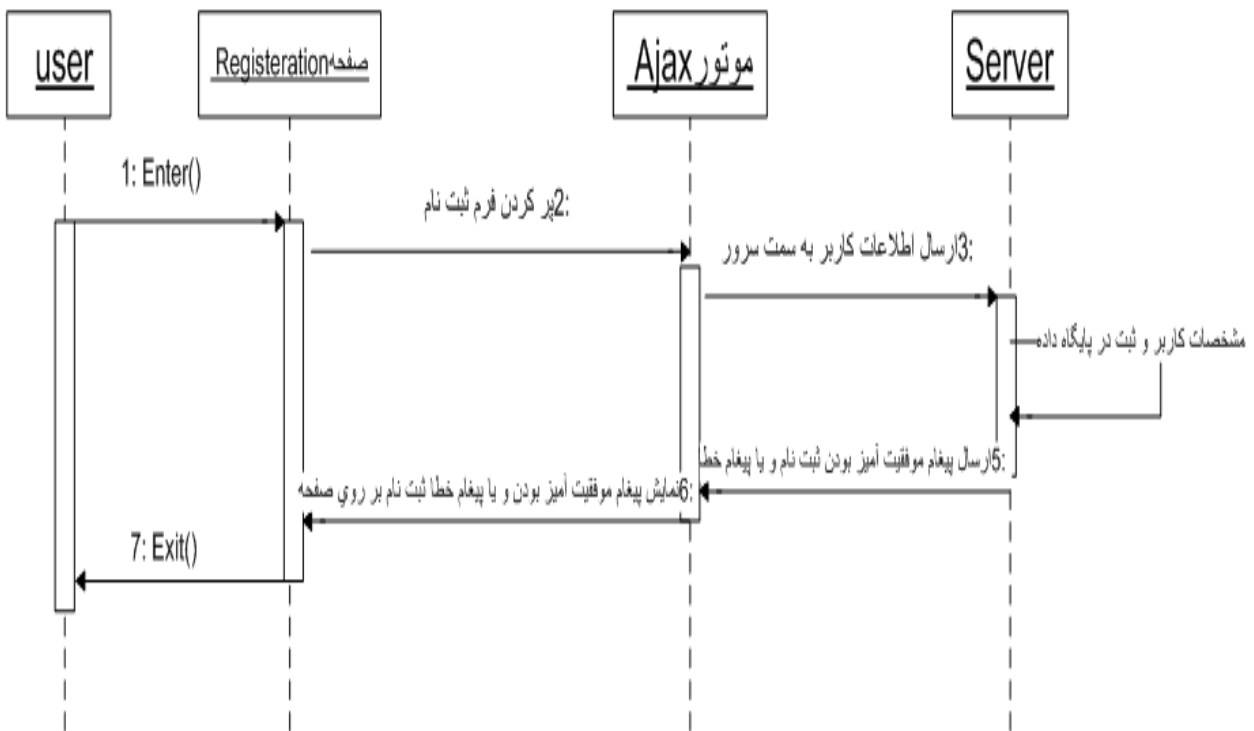
ادامه نمودار سطح صفر DFD:



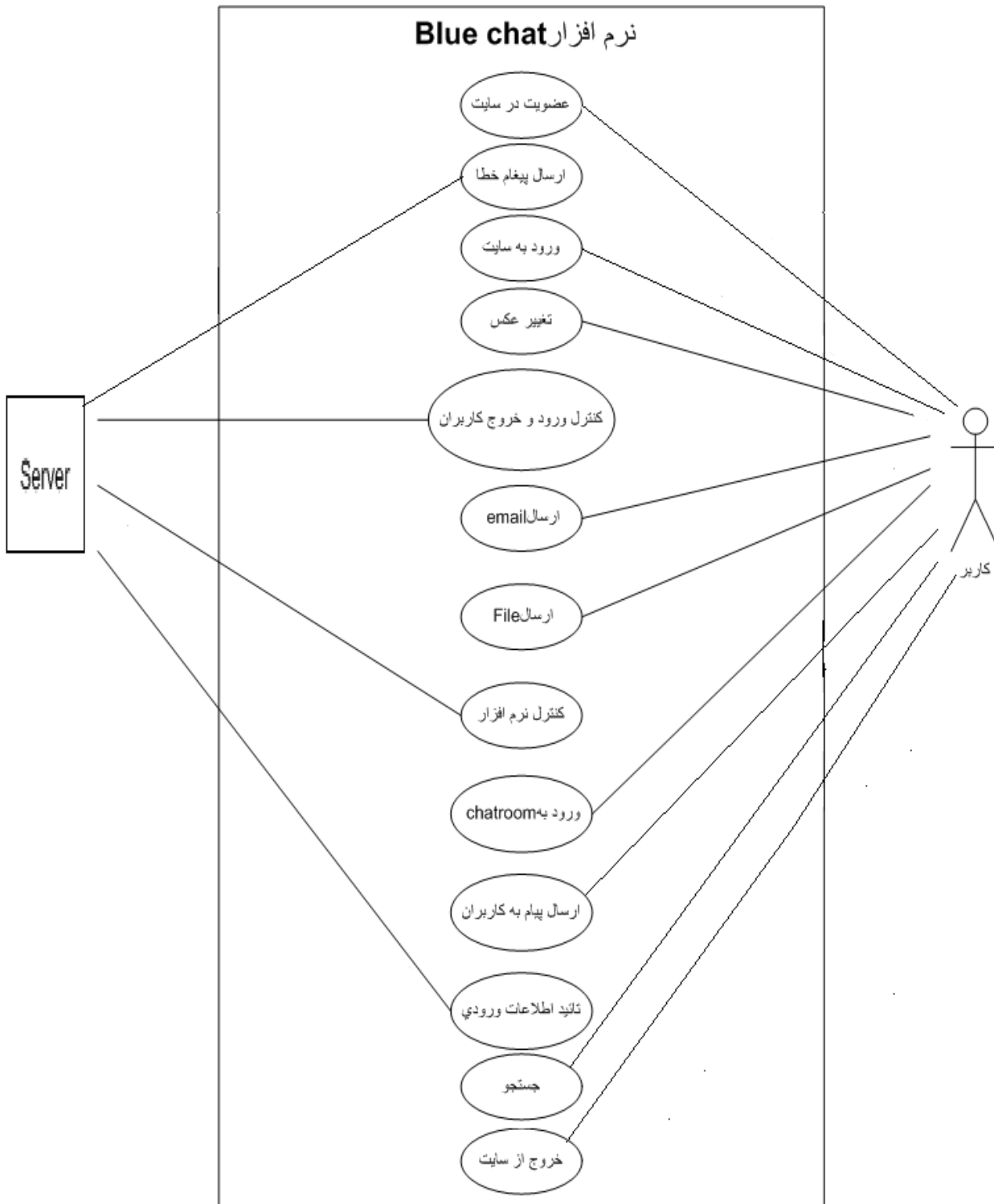
دیاگرام ترتیب برای ورود اعضا به سایت:



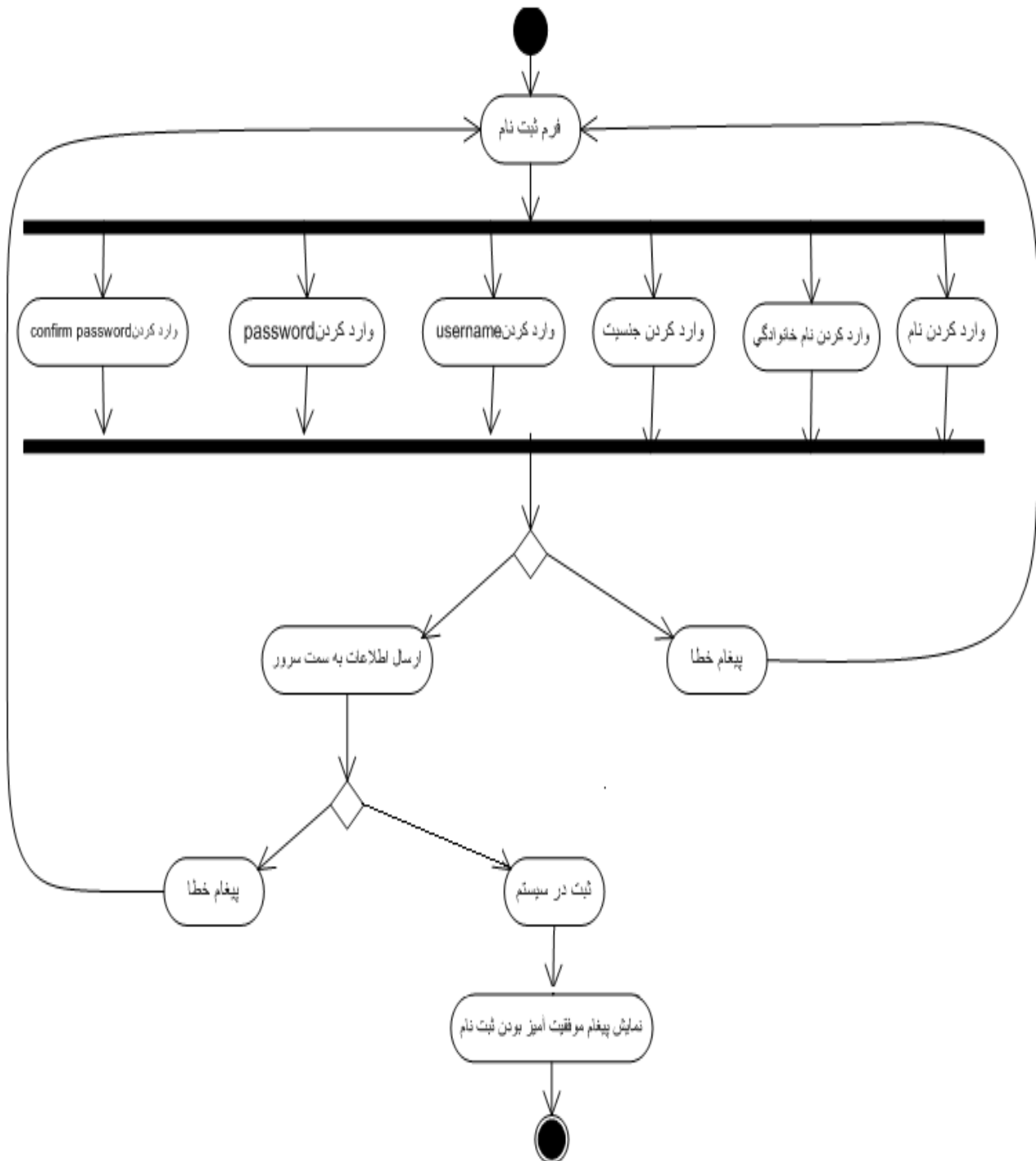
دیاگرام ترتیب برای ثبت کاربر جدید:



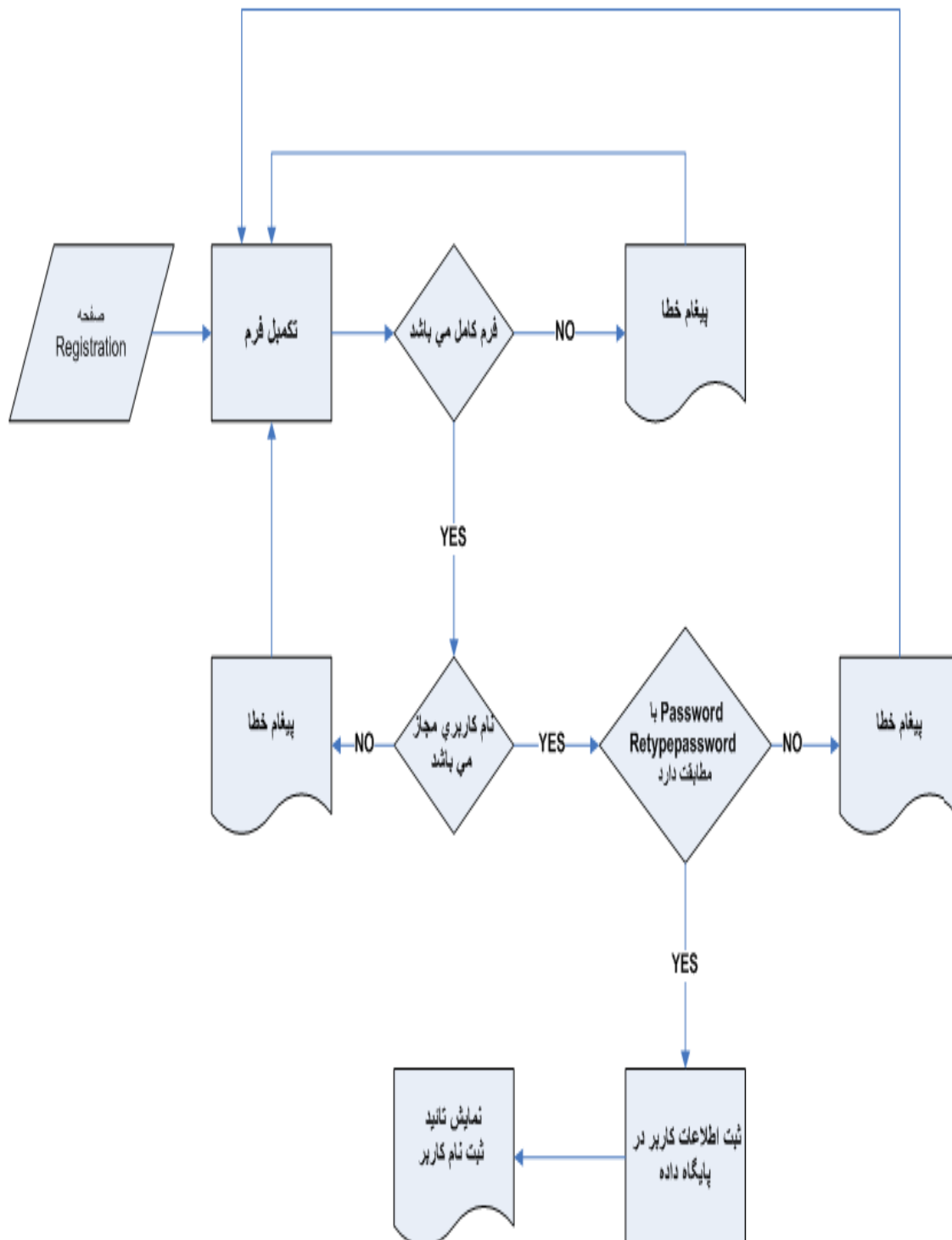
دیاگرام UseCase برای روند کلی سیستم:



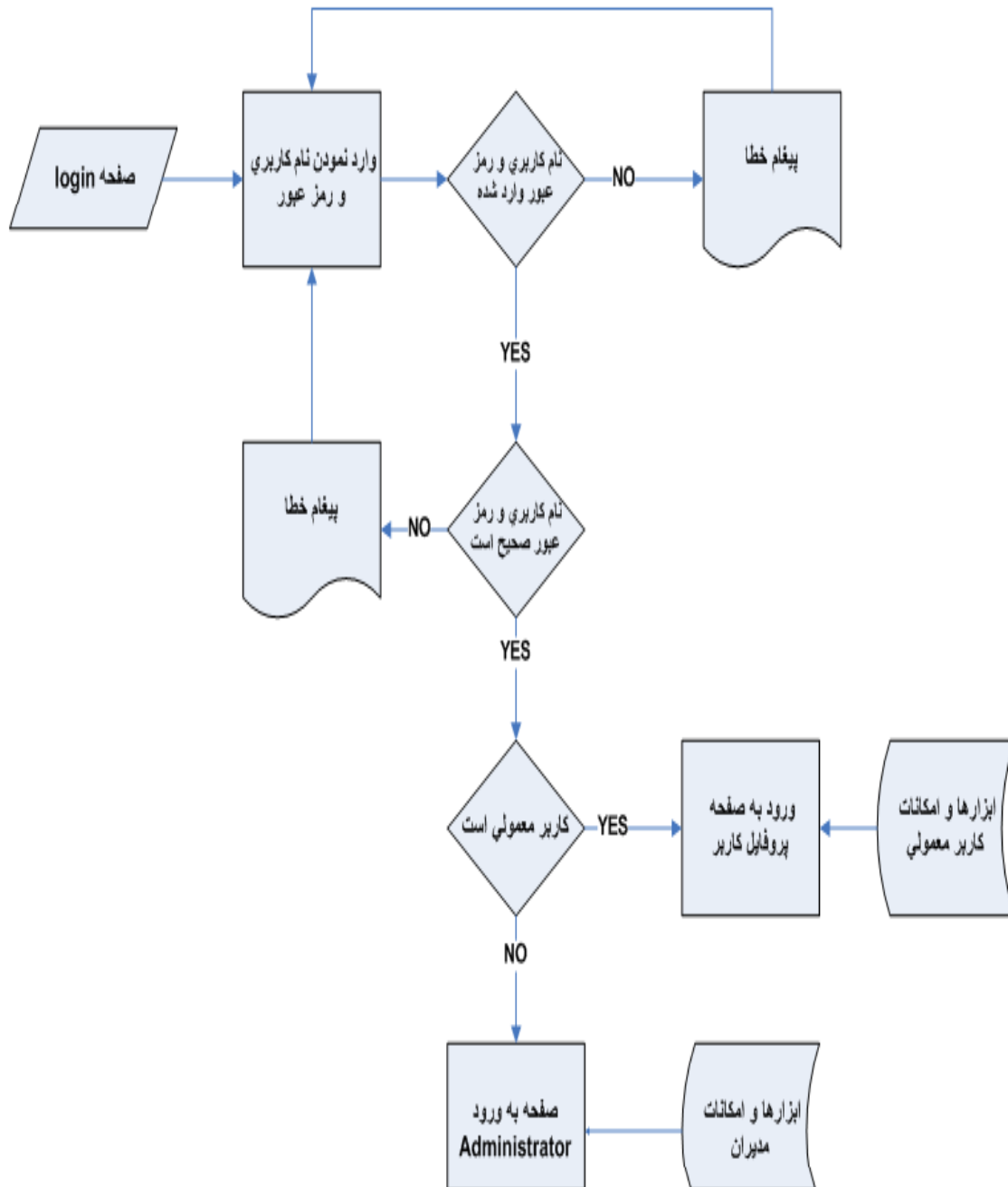
دیاگرام فعالیت برای روند ثبت نام:



روند نمای ثبت نام کاربر:



روند نمای ورود اعضا:



فصل سوم: مستندات پیاده سازی سیستم

۳-۱- نرم افزارهای مورد استفاده

در این پروژه برای طراحی و پیاده سازی سایت از نرم افزارهای زیر استفاده شده است.

- آجاکس
- VS.NET 2005
- ASP.NET(C#)
- Photoshop 8

۳-۲- کارآیی و امکانات سایت:

در این پروژه هدف اصلی طراحی نرم افزار وب چت^۱ است که مبتنی بر آجاکس می باشد. علاوه بر صفحات طراحی شده یک فایل کتابخانه ای به نام ChatControl.dll با استفاده از جاوااسکریپت^۲ و سی شارپ^۳ طراحی شده است که در این فایل کتابخانه ای

¹ Web Chat

² Javascript

³ C#

قسمت چت روم^۱ سایت طراحی شده و در کد جاوااسکریپت آن از کدهای آجاکس استفاده شده است. در ادامه به توضیح هر یک از امکانات سایت خواهیم پرداخت.

۳-۲-۱- صفحات طراحی شده:

در این قسمت به شرح و بررسی صفحات طراحی شده می‌پردازیم:

۳-۲-۱-۱- صفحه Login.aspx:

این صفحه اولین صفحه‌ای است که پیش روی کاربر قرار می‌گیرد. کاربر می‌تواند پس از عبور از این صفحه به پروفایل شخصی خود دسترسی پیدا کند. این صفحه دارای امکانات زیر می‌باشد:

- اطلاعات کلی در مورد سایت
- Demo Blue Chat: کاربران می‌توانند با کلیک کردن بر روی این لینک فایل Demo Blue Chat را بر روی سیستم خود نصب کنند. این فایل با نرم افزار فلش طراحی شده است و قسمت‌های مختلف سایت را به کاربر نمایش می‌دهد.
- Get e new ID: کاربران می‌توانند با کلیک کردن بر روی این لینک وارد صفحه ثبت نام^۲ شوند و در سایت ثبت نام نمایند.
- فرم مخصوص ورود به سیستم^۳: اعضا می‌توانند با کامل کردن این سایت پروفایل اصلی خود وارد شوند. در این فرم اعضا می‌بایست نام کاربری^۴ و کلمه رمز^۵ خود را به صورت صحیح وارد نمایند و سپس بر روی دکمه login کلیک نمایند، در صورت نبودن هیچ گونه خطایی کاربر وارد صفحه پروفایل^۶ می‌شود.

پیغام‌های کنترل خطادر صفحه login:

در صفحه login، ۲ نوع پیغام خطا در نظر گرفته شده که تمامی خطاهایی که ممکن است از کاربر سر بزند را کنترل می‌کند.

¹ ChatRoom
² Registretion
³ Login
⁴ Username
⁵ Password
⁶ Profile

- **خطای خالی بودن جعبه متن^۱:** این کنترل با استفاده از ابزار `TextboxWatermarkExtender` نوشته شده که یکی از ابزارهای آجاکس است. با استفاده از این ابزار در صورت خالی بودن جعبه متن پیغامی به کاربر نمایش داده می‌شود تا کاربر از خالی بودن جعبه متن مطلع شود و اقدام به پر نمودن آن نماید.
- **خطای اشتباه بودن نام کاربری و یا کلمه رمز:** این خطا زمانی اتفاق می‌افتد که کاربر فیلدهای نام کاربری و یا کلمه رمز را به صورت ناصحیح پر نموده‌است. در این پیغام از کاربر خواسته می‌شود که آنها را به صورت صحیح وارد نماید.

تابع `login_click()`:

در صفحه `login` وظیفه اصلی بر عهده دکمه `login` می‌باشد و این دکمه پس از اینکه بر روی آن کلیک شد اعمال زیر را به ترتیب انجام می‌دهد.

- (۱) در صورت خالی نبودن فیلدهای مربوط به نام کاربری و کلمه رمز اعمال زیر را به ترتیب انجام می‌دهد.
- (۲) در جدول ثبت^۲ سطری را جستجو می‌کند که نام کاربری و کلمه رمز آن با نام کاربری و کلمه رمزی که کاربر وارد نموده‌است مطابقت داشته باشد، در صورت یافتن چنین سطری وارد مرحله بعدی می‌شود و در غیر این صورت پیغام خطایی را برای کاربر نمایش می‌دهد.
- (۳) در این مرحله صفحه `profile.aspx` را فراخوانی کرده و همچنین نام کاربری را به این صفحه پاس می‌دهد.

۳-۲-۱-۲-۳ صفحه `Registretion.aspx`:

صفحه ثبت نام^۳ به این منظور طراحی شده‌است که کاربران بتوانند عضو سایت شوند و از امکانات آن استفاده کنند. در این صفحه سعی شده‌است مانند دیگر صفحات سایت به منظور کاهش مدت انتظار کاربر در حین ثبت نام از تکنولوژی آجاکس استفاده شود. امکانات این صفحه به شرح زیر است:

- اطلاعات کلی در مورد سایت

^۱ Textbox

^۲ Register

- **View my profile**: در صورت موفقیت آمیز بودن ثبت نام کاربر، با کلیک کردن بر روی این لینک کاربر به صفحه login وارد شده و پس از وارد کردن نام کاربری و کلمه رمز خود می تواند پروفایل خود را مشاهده کند.
 - فرم مخصوص ثبت نام: در این صفحه یک فرم ثبت نام پیش روی کاربر قرار می گیرد که پرکردن آن برای ثبت نام در سایت ضروری است. اولین قسمت در نظر گرفته شده برای این فرم اطلاعات شخصی کاربر از قبیل نام و نام خانوادگی، نوع جنسیت و... است. در قسمت بعدی از کاربر تقاضا می شود که برای خود نام کاربری انتخاب کند که برای هر کاربر به صورت منحصر به فرد است و فرمت آن باید به صورت e.g@bluechat.com باشد. پس از آن کاربر کلمه رمز خود را وارد می کند. پس از اینکه کاربر اطلاعات مورد نیاز را وارد نمود با فشردن دکمه Register عملیات اصلی ثبت نام که بر عهده تابع Register_click() است، آغاز می شود.
- قبل از اینکه به بررسی تابع Register_click() بپردازیم لازم است توضیحی در مورد پیام های کنترلی که در این صفحه در نظر گرفته شده است، داشته باشیم.

پیغامهای کنترلی در صفحه Registration.aspx:

به طور کلی در این صفحه ۵ پیام کنترلی در نظر گرفته شده است:

- (۱) **خطای خالی بودن فیلد**: به منظور جلوگیری از خطای خالی بودن فیلد با استفاده از ابزار Textboxwatermark که یکی از ابزارهای آجاکس است این خطا را کنترل می کنیم. عملکرد این ابزار به این صورت است که قبل از اینکه صفحه به سمت سرور فرستاده شود به صورت هوشمند از خالی بودن فیلد باخبر شده و به کاربر پیغام خالی بودن فیلد را می دهد و از کاربر تقاضا می کند که فیلد مورد نظر را پر نماید.
- (۲) **خطای نام کاربری**: این خطا زمانی اتفاق می افتد که کاربر نام کاربری تکراری را وارد نماید. برای مطلع کردن کاربر از این خطا پیغامی را نمایش می دهد.
- (۳) **خطای کلمه رمز**: زمان وقوع این خطا هنگامی است که تعداد کاراکترهای کلمه رمز کمتر از ۷ باشد، زیرا حداقل طول کلمه رمز ۷ در نظر گرفته شده و کلمه رمز باید حداقل ۷ کاراکتر داشته باشد.
- (۴) **خطای مربوط به تایپ مجدد کلمه عبور^۱**: در صورت عدم تطابق کلمه رمز با تایپ مجدد آن این خطا اتفاق می افتد. برای کنترل خطاهای شماره ۳ و ۴ از ابزار CustomValidation استفاده می شود.

¹ Retype Password

۵) پیغام مربوط به موفقیت آمیز بودن ثبت نام: این پیغام زمانی که عملیات ثبت نام کاربر با موفقیت به اتمام رسیده باشد به کاربر نمایش داده می‌شود.

تابع Register_Click():

وظیفه اصلی ثبت نام بر عهده این تابع است و هر بار که کاربر دکمه Register را می‌فشارد این تابع اعمال زیر را به ترتیب انجام می‌دهد.

۲) در ابتدا خطاهای ۱، ۳ و ۴ را کنترل نموده و در صورت نبودن هیچ گونه خطایی وارد مرحله بعدی می‌شود.

۲) در این مرحله به بررسی خطای شماره ۲ می‌پردازد. به همین منظور در جدول ثبت سطری را جستجو می‌کند که نام فیلد نام کاربری آن با نامی که کاربر وارد نموده‌است برابر باشد. در صورت نیافتن چنین فیلدی پیغام خطای ۲ را به کاربر نمایش می‌دهد و در صورت عدم وجود خطا اقدام به انجام مرحله بعدی می‌کند.

۳) در این مرحله سعی می‌شود تا اطلاعات وارد شده توسط شخص در جدول Register، ثبت شود. پس از اتمام این مرحله پیغامی نمایش داده می‌شود تا کاربر از اتمام ثبت نام خود مطلع شود.

یکی از ویژگی‌های مهم این صفحه استفاده از ابزار قدرتمند updatepanel است. این ابزار همانطور که قبلاً هم گفته شده از ابزارهای آجاکس است. این ابزار باعث بالا رفتن سرعت عملیات ثبت نام می‌شود و مانع از بارگذاری^۱ مجدد صفحه پس از اولین بارگذاری می‌شود. به علاوه این ابزار باعث حذف شدن عملیات توقف^۲، حرکت به جلو^۳، بازگشت^۴ در صفحه IE می‌شود، چون صفحه یکبار بارگذاری می‌شود.

این ابزار باعث می‌شود که تنها اطلاعات مورد نیاز از قبیل نام، نام خانوادگی و... به سمت سرور فرستاده می‌شود و مانع از ارسال کل صفحه به سمت سرور می‌شود. در طرف سرور نیز در هنگام پاسخگویی به تقاضا تنها اطلاعات مورد نیاز از قبیل پیغام‌های خطا و پیغام‌های تأیید به سمت صفحه فرستاده می‌شود.

۳-۲-۱-۳- صفحه Profile.aspx:

¹ Load

² Stop

³ Forward

⁴ Back

این صفحه پر محتواترین صفحه سایت است و در واقع بیشترین امکانات سایت در این صفحه قرار دارد و خود شامل زیر صفحاتی است که در Tab Container قرار دارند. لازم به ذکر است که در طراحی صفحاتی که غالب یکسانی دارند و تنها در یک قسمت از صفحه با هم متفاوت هستند شما می‌توانید از روش‌های متفاوتی استفاده کنید تا غالب تمامی آنها به صورت یکسان و مشابه باشد. یکی از این روش‌ها استفاده از صفحه اصلی^۱ است و روش دیگری که آجاکس پیش روی ما می‌گذارد و روش سریع‌تری نسبت به صفحه محتوی می‌باشد، استفاده از ابزار TabContainer است. ما در طراحی این سایت از این ابزار برای ساختن زیر صفحات استفاده کرده ایم که در ادامه به توضیح هر یک از این زیر صفحات خواهیم پرداخت.

امکانات صفحه پروفایل:

✓ لینک‌ها:

• لینک ویرایش پروفایل^۲

• لینک چت روم^۳

• لینک اطلاعات کاربر^۴

در ادامه به توضیح این لینک‌ها خواهیم پرداخت.

✓ برچسب‌ها (تب^۵):

• برچسب ارسال نامه الکترونیکی

• برچسب دریافت نامه الکترونیکی

• برچسب ارسال فایل

• برچسب دریافت فایل

• برچسب جستجو

• برچسب اضافه و حذف کردن دوست به لیست

¹ Master Page

² Edit Profile link

³ Chat Room link

⁴ User Information link

⁵ Tab

• برجسب دریافت دعوت نامه

در ادامه نیز به توضیح این برجسبها می پردازیم.

بررسی عملکرد وامکانات هر یک از برجسبهای صفحه پروفایل:

۷ برجسب ارسال نامه الکترونیکی:

یکی از مهمترین امکاناتی که در صفحه پروفایل در اختیار کاربر قرار می گیرد، ارسال نامه الکترونیکی است. کاربر با انتخاب این برجسب صفحه مربوط به ارسال نامه الکترونیکی را مشاهده خواهد کرد. در این صفحه کاربر می تواند با پر کردن فرم نامه و پس از آن با فشردن دکمه Send نامه خود را ارسال کند. در فرم نامه علاوه بر متن کاربر می تواند با انتخاب دکمه Attach فایلی را برای ارسال انتخاب کند.

در برجسب مخصوص ارسال نامه الکترونیکی هر یک از دکمه‌ها نقشی را ایفا می کنند، در این قسمت به توضیح آنها می پردازیم:

- دکمه Browes:

این دکمه به کاربر اجازه می دهد تا فایلی را از روی سیستم خود انتخاب کرده و آن را به عنوان پیوست¹ به نامه الحاق کند.

- دکمه Reset:

چنانچه کاربر از ارسال نامه منصرف شده باشد و یا بنا به دلایلی بخواهد فرم نامه را پاک کند، می تواند از این دکمه استفاده نماید.

- دکمه Send:

مهمترین دکمه‌ای که در این صفحه قرار دارد، دکمه Send است. این دکمه پس از اینکه توسط کاربر فشرده شد، جهت ارسال نامه الکترونیکی مراحل زیر را می پیماید.

مرحله ۱:

ابتدا تمامی خطاهایی که ممکن است از کاربر سر بزند را چک می کند و در صورت عدم وجود خطا وارد مرحله بعد می شود.

مرحله ۲:

در این مرحله چک می کند که آیا کاربر فایلی را به عنوان پیوست انتخاب کرده است یا خیر. اگر انتخاب نکرده باشد وارد مرحله بعد می شود و در غیر این صورت پسوند فایل را در یک متغیر به نام Passfile ذخیره می کند و سپس وارد مرحله بعد می شود.

¹ Attachment

مرحله ۳:

در این مرحله با برقراری ارتباط با پایگاه داده تمامی اطلاعات فرم نامه را به همراه نام فرستنده در جدول Mail ذخیره می‌کند. علاوه بر این می‌بایست فایلی را که به عنوان پیوست ارسال شده‌است را در سمت سرور ذخیره کند. به منظور جلوگیری از ذخیره فایل‌هایی با نام تکراری به هر نامه الکترونیکی^۱ یک IdEmail که عددی منحصر به فرد است اختصاص داده‌ایم. در نتیجه فایل مذکور با نام IdEmail+Passfile در سمت سرور و در پوشه Attachfile ذخیره می‌شود.

مرحله ۴

در این مرحله پیام "Your Email have been send" برای کاربر نمایش داده می‌شود.

خطاهای کاربر در هنگام ارسال نامه الکترونیکی:

- خطای خالی بودن فیلد گیرنده نامه:

در صورت خالی بودن فیلد گیرنده نامه هیچ اطلاعاتی سمت سرور فرستاده نمی‌شود و همواره از کاربر خواسته می‌شود که این فیلد را کامل نماید. این کنترل با استفاده از TextBoxWatermark که یکی از ابزارهای آجاکس است نوشته شده‌است.

- خطای بزرگ بودن ساینز فایل ارسالی:

حد اکثر ساینز که برای فایل پیوست در نظر گرفته شده‌است ۱ مگا بایت است، و در صورت بزرگ تر بودن فایل از مقدار مجاز به کاربر پیغام داده می‌شود.

✓ برچسب دریافت نامه الکترونیکی:

هنگامی که صفحه پروفایل کاربر برای اولین بار، بارگزاری می‌شود. در تابع Page_load این صفحه با اتصال به بانک اطلاعاتی و جستجو در جدول Mail در صورت یافتن نامه الکترونیکی جدید پیغامی در قسمت پیام خطا^۲ این صفحه به کاربر نمایش داده می‌شود و این پیغام حاوی این مضمون است که کاربر دارای نامه جدید است. حال کاربر می‌تواند به قسمت Check Mail مراجعه کرده و mailهای خود را دریافت کند.

در برچسب نامه الکترونیکی تمامی نامه‌هایی را که کاربر تا اکنون دریافت کرده‌است نمایش داده می‌شود.

¹ Email

² Alert

این نامه‌ها با توجه به تاریخ ارسال مرتب شده اند و هر یک در صفحه ای مجزا نمایش داده می‌شوند.

به منظور نمایش نامه‌ها از ابزار Formview استفاده شده‌است، البته در ظاهر فرم تغییراتی داده ایم برای مثال برای نمایش متن نامه به جای برچسب^۱ از جعبه متن^۲ که قابلیت حرکت متن^۳ را دارد استفاده شده‌است، و در قسمت پیوست فرم از ابزار Hyperlink استفاده کرده‌ایم تا کاربر با کلیک کردن بر روی آن بتواند فایلی را که برای او ارسال شده‌است در کامپیوتر شخصی خود ذخیره^۴ و یا باز^۵ کند.

برای اینکه FormView نامه‌های کاربر خاص را نمایش دهد. می‌بایست هنگامی که شما آن را به بانک اطلاعاتی متصل می‌کنید شرط مربوط به Where را Username=@lbUsername قرار دهید. که lbUsername یک برچسب در صفحه پروفایل است که هنگامی که صفحه پروفایل بارگذاری می‌شود این برچسب مقدار گرفته و مقدارش را برای FormView ارسال می‌کند.

✓ برچسب جستجو:

در این بخش از صفحه پروفایل کاربران می‌توانند با وارد نمودن نام کاربری و یا نام شخص مورد نظر اطلاعات مربوطه را مشاهده کنند. در این بخش وظیفه اصلی بر عهده دکمه جستجو است. هنگامی که کاربر این دکمه را می‌فشارد، این دکمه کار خود را آغاز می‌کند. ابتدا به بانک اطلاعاتی متصل می‌شود سپس سطری را که دارای یکی از دو شرط زیر باشد جستجو می‌کند.

Username = [نام کاربری مورد نظر]

FirstName = [نام مورد نظر]

پس از آن تمامی سطرهای یافت شده را در DataGridView نمایش می‌دهد.

برای اینکه DataGridView فقط سطرهای خاصی از جدول را نمایش دهد می‌باید هنگام متصل کردن آن به بانک

اطلاعاتی شرط Where را طوری تنظیم کنید که مقدار UserName را از جعبه متن موجود در صفحه بگیرد. در صورتی که سطر فوق یافت نشود پیغام "Not found any result" نمایش داده می‌شود.

پیغام‌های کنترلی در هنگام جستجو دوستان:

¹ Lable

² TextBox

³ Scroll

⁴ Save

⁵ Open

- پیغام پر کردن فیلد مربوط به نام کاربری در صورت خالی بودن این فیلد
- پیغام یافت نشدن نتیجه

✓ برچسب حذف و اضافه کردن دوست به لیست:

این برچسب دارای دو قسمت مجزا می باشد. یک قسمت مربوط به اضافه کردن دوست به لیست و دیگری مربوط به حذف دوست از لیست است.

اضافه کردن دوست به لیست کاربر:

در این قسمت کاربر می بایست نام کاربری کاربری را که می خواهد به عنوان دوست به لیست خود اضافه کند را وارد نموده، سپس بر روی دکمه Add کلیک کند. پس از اینکه کاربر بر روی Add کلیک کرد این دکمه عملیات زیر را به ترتیب انجام می دهد.

مرحله ۱) بررسی خطاهایی که ممکن است از کاربر سر بزند و در صورت عدم وجود خطا وارد مرحله بعد می شود.

مرحله ۲) تمامی اطلاعات مربوط به فرم اضافه کردن دوست را به همراه نام کاربری شخص تقاضا دهنده در جدولی به نام Friend ثبت می کند. در جدول Friend فیلدی به نام Sfriend وجود دارد که وضعیت پاسخ دهی به دعوت نامه را مشخص می کند. چنانچه مقدار این فیلد صفر باشد یعنی دعوتنامه برای گیرنده ارسال شده است اما گیرنده هنوز پاسخ نداده است. و اگر مقدارش ۱ باشد یعنی گیرنده به دعوتنامه جواب مثبت داده است. لذا هنگامی که کاربری دعوتنامه ای برای کاربری ارسال می کند، مقدار این فیلد در جدول صفر در نظر گرفته می شود.

لازم به ذکر است تا زمانی که گیرنده به دعوت نامه پاسخ مثبت نداده است نام آن در لیست دوستان فرستنده ظاهر نمی شود و فرستنده همچنان می بایست منتظر پاسخ بماند.

مرحله ۳) پیغام "Your Invitation have been send" نمایش داده می شود.

خطاهای کاربر در هنگام اضافه کردن دوست به لیست:

- خطا خالی بودن فیلد مربوط به نام کاربری

- **عدم عضویت گیرنده دعوت نامه در سایت:** برای کنترل این خطا پس از فشردن دکمه Send، با اتصال به بانک اطلاعاتی و جستجو در جدول Register در صورت عدم یافتن سطری که نام کاربری آن با نام کاربری گیرنده برابر باشد، این خطا گزارش می شود.

حذف دوست از لیست:

در این قسمت کاربر نام شخص مورد نظر را وارد نموده و در صورت عدم وجود خطا با فشردن دکمه Delete این شخص از لیست دوستانش حذف می شود.

بررسی عملکرد دکمه Delete:

این دکمه وظیفه دارد با متصل شدن به بانک اطلاعاتی سطری را که دارای هر دو شرط زیر است جستجو کند.

Friend1 = [نام کاربر تقاضا دهنده]

Friend2 = [نام کاربری که می خواهیم از لیست حذف شود]

چنانچه چنین سطری را یافت آن را حذف می کند و به کاربر پیغام "This user delete from your list" را نمایش می دهد. و چنانچه این سطر یافت نشود به کاربر پیغام می دهد که چنین کاربری یافت نشد لطفا دوباره سعی کنید.

خطاهای کاربر در هنگام حذف:

- **خطا خالی بودن فیلد مربوط به نام کاربری**

- **عدم وجود نام کاربری در لیست دوستان:** این خطا هنگامی رخ می دهد که نام کاربری که می خواهیم آن را حذف کنیم در لیست دوستانمان نباشد.

✓ برچسب ارسال فایل:

این تب مخصوص ارسال فایل طراحی شده و به کاربر اجازه می دهد تا برای دوستان خود فایل ارسال نماید. در این قسمت ابتدا نام کاربری را که می خواهد فایل را برای آن ارسال نماید در کنترل Txtusername وارد می نماید. سپس با کلیک کردن بر روی دکمه ی browse آدرس فایل مورد نظر را انتخاب می کند. پس از آن با کلیک بر روی کلمه ارسال، فایل را ارسال می نماید.

بررسی عملکرد دکمه send file:

(۱) خطایی که ممکن است از کاربر سر بزند را کنترل می کند و در صورت عدم وجود خطا وارد مرحله ی بعد می شود.

۲) پسوند فایل ارسالی را جدا نموده و آن را در یک متغیر به نام pass file2 ذخیره می‌کند.

۳) تمامی اطلاعات از قبیل نام کاربر فرستنده، نام کاربر گیرنده، زمان ارسال، نام فایل ارسالی، سایز فایل ارسالی و... را در جدول Files ذخیره می‌کند.

به علاوه فایل ارسالی را در سمت سرور در پوشه ای به نام فایل ذخیره می‌کند. به منظور جلوگیری از ذخیره شدن فایل‌ها با نام مشابه، در جدول Files فیلدی به نام ID file در نظر گرفتیم که این شناسه^۱ منحصر به فرد است. هنگامی که می‌خواهیم فایل ارسالی را در سمت سرور ذخیره کنیم، نام آن را به ID file+pass file2 تغییر می‌دهیم و سپس آن را با این نام جدید ذخیره می‌کنیم.

پیغام‌های کنترلی در هنگام ارسال فایل:

- پیغام پر نمودن فیلد مربوط به گیرنده
- پیغام خالی بودن آدرس فایل ارسالی
- خطای عدم وجود گیرنده در لیست اعضاء سایت

۴ برچسب دریافت فایل:

یکی دیگر از پیغام‌های هشداری که ممکن است بر روی صفحه پروفایل کاربر در بخش پیام خطا به چشم بخورد پیغام مربوط به دریافت فایل جدید است. همانند دریافت نامه الکترونیکی هنگامی که صفحه پروفایل کاربر برای اولین بارگزاری می‌شود در تابع page-load با اتصال به بانک اطلاعاتی و جستجو در جدول Files در صورت یافتن نامه الکترونیکی جدید، پیغامی در قسمت پیام خطا صفحه پروفایل به کاربر نمایش داده می‌شود. این پیغام حاوی این مضمون است که کاربر دارای n فایل دریافت کرده است. حال کاربر می‌تواند به قسمت check file مراجعه کرده و فایل‌های خود را مشاهده کند.

برای نمایش تمامی فایل‌های کاربر خاص ابتدا می‌بایست به بانک اطلاعاتی متصل شویم سپس در جدول Files سطرهایی را جستجو می‌کنیم که فیلد receiver file آن برابر با نام کاربر باشد. در صورت یافتن حداقل یک سطر با استفاده از کدهای زیر جدولی پویا ایجاد می‌کنیم و تمام اطلاعات مربوط را در این جدول قرار می‌دهیم. در یکی از ستون‌های این جدول hyperlink قرار داده‌ایم و هنگامی که کاربر بر روی آن کلیک می‌کند می‌تواند فایل مورد نظر را بر روی کامپیوتر شخصی خود قرار داده‌ایم و یا باز نماید. چنانچه کاربر هیچ فایلی را تاکنون دریافت نکرده باشد پیغامی نمایش داده می‌شود.

¹ ID

✓ برچسب دریافت دعوت نامه:

سومین پیامی که در صفحه ی پروفایل در قسمت پیام خطا به کاربر نمایش داده می شود، پیام مربوط به دریافت دعوت نامه است. کاربر با دریافت این پیام متوجه می شود که می بایست وارد تب مربوط به دعوت نامه شود و در این قسمت به دعوت نامه های خود پاسخ می دهد.

فرم مربوط به دعوت نامه شامل نام کاربر فرستنده، زمان ارسال دعوت نامه، متن ارسالی از طرف فرستنده، لینک view friend profile و دو دکمه با نام های yes و no است. کاربر می تواند با کلیک کردن بر روی لینک view friend profile، وارد صفحه user Information شود و اطلاعات مربوط به شخص فرستنده دعوت نامه را ملاحظه کند. گیرنده دعوت نامه می بایست با کلیک کردن به روی یکی از دو دکمه ی yes و no نظر خود را در مورد این دوستی اعلام کند.

بررسی عملکرد yes:

همانطور که در بخش اضافه کردن دوست به لیست نیز گفته شد، هنگامی که کاربر پاسخ مثبت به دعوت نامه دهد در جدول friend و در سطر مربوط به این دعوت نامه فیلد Sfriend مقدار ۱ می گیرد.

سطر دیگری به جدول friend اضافه می شود که فیلد [گیرنده]=friend1 و [فرستنده]=friend2 و Sfriend=1 است. در واقع با اضافه کردن این سطر به جدول شخص فرستنده نیز به لیست دوستان شخص گیرنده اضافه می شود و این دوستی یک دوستی دوطرفه خواهد بود.

بررسی عملکرد no:

هنگامی که کاربر به دعوت نامه جواب منفی می دهد، سطر مربوط به این دعوت نامه در جدول friend پاک خواهد شد.

بررسی لینک های صفحه پروفایل:

❖ User Information

در این قسمت ابتدا نام کاربری را که شما می خواهید اطلاعاتش را مشاهده بفرمائید، خواسته می شود سپس با کلیک کردن بر روی View user information شما وارد صفحه User Information می شوید. عملکرد این لینک به این صورت است که هنگامی شما بر روی آن کلیک می کنید در جدول Register به دنبال کاربری با این نام می گردد و اگر آن را نیافت در بالای صفحه پروفایل پیغامی با این مضمون که چنین کاربری یافت نشد نمایش می دهد. اما در صورت یافت

شدن نام کاربری این کاربر را برای صفحه User Information ارسال می کند. در صفحه User Information و در تابع Page_load با دریافت نام کاربری کاربر در جدول Register به دنبال چنین کاربری می گردد و تمامی اطلاعات آن را در صفحه و در مکان های مناسب می دهد. برای درک بهتر این روند به قطعه کدهای زیر توجه فرمائید.

❖ **Edit Profile**: این قسمت به این منظور طراحی شده تا کاربر بتواند اطلاعات شخصی خود را ویرایش کند. کاربر با کلیک کردن بر روی این لینک وارد صفحه Edit Profile می شود.

❖ **Chat Room**: مهم ترین قسمت این نرم افزار، صفحه Chatroom می باشد. کاربر می تواند با کلیک کردن بر روی این لینک وارد صفحه Chatroom شود. قبل از اینکه کاربر وارد صفحه Chatroom شود اطلاعات آن در جدول Userinchat ذخیره می شود. در صفحه Chatroom در تابع Pag_load با اتصال به بانک اطلاعاتی نام تمامی افراد موجود در جدول Userinchat را در قسمت User online صفحه Chatroom نمایش می دهد.

۳-۲-۱-۴- صفحه Edit Profile:

کاربران با وارد شدن به این صفحه می توانند پروفایل و اطلاعات خود را ویرایش کنند.

امکانات صفحه Edit Profile:

(۱) اختصاصی بودن این صفحه، به این منظور که هر کاربر صفحه Edit Profile خود را دارد و یک کاربر نمی تواند اطلاعات کاربر دیگر را تغییر دهد.

(۲) ویرایش و یا تکمیل نمودن اطلاعات شخصی از قبیل نام، نام خانوادگی، کشور، تاریخ تولد، جنسیت

(۳) ویرایش رمز عبور

(۴) ویرایش عکس

بررسی عملکرد توابع موجود در صفحه EditProfile.aspx.Cs:

تابع Change personal info():

این تابع که در رویداد کلیک دکمه Change info (این دکمه در تب Personal Information قرار دارد) نوشته شده است، وظیفه دارد تا با اتصال به بانک اطلاعاتی تمامی اطلاعات جدیدی را که کاربر وارد نموده است در جدول Register وارد نماید. برای این منظور از تابع Update در Sql استفاده کرده است.

تابع (Change password):

این تابع در رویداد کلیک دکمه Change Password (این دکمه در تب Password قرار دارد) نوشته شده است، وظیفه دارد که کارهای زیر را به ترتیب انجام دهد:

۱) در ابتدا چک می کند که previous password با آنچه در جدول Register وجود دارد مطابقت داشته باشد و در صورت عدم تطابق پیغام Not verify previous password به کاربر نمایش داده می شود. در صورت تطابق داشتن وارد مرحله بعد می شود.

۲) در این مرحله چک می کند که New password با Retype New Password مطابقت داشته باشد. در صورت عدم تطابق پیغام خطا می دهد. در صورت تطابق داشتن وارد مرحله بعد می شود.

۳) پسورد را در جدول Register و در سطری که نام کاربری آن با کاربری که می خواهد پروفایلش را ویرایش کند برابر باشد، ثبت می نماید.

تابع (Change Image):

این تابع در رویداد کلیک دکمه Change Image (این دکمه در تب Image قرار دارد) نوشته شده است و وظیفه تغییر عکس را بر عهده دارد. در تب مربوط به عکس^۱ هنگامی که کاربر از طریق جستجو^۲ تصویر مورد نظر خود را انتخاب کرد و سپس بر روی دکمه Change Image کلیک می کند.

۳-۲-۱-۵- صفحه User Information:

این صفحه اطلاعات کاربر خاصی را نمایش می دهد.

بررسی تابع (Page_load):

در این تابع ابتدا چک می شود که آیا این صفحه توسط دیگر صفحات فراخوانده شده است. در صورت برقرار شدن شرط نام کاربری به این صفحه پاس داده می شود و در متغیری به نام uname ذخیره می کند. سپس در جدول Register به دنبال سطری می گردد که فیلد نام کاربری آن با uname برابر باشد. به محض یافتن چنین سطری اطلاعات آن سطر را بر روی صفحه و در مکان خاص نمایش می دهد.

¹ Image

² Browse

در این تابع برای نمایش دادن عکس کاربر مربوطه ابتدا فیلد SImage موجود در جدول Register را چک می‌کند. اگر مقدار این فیلد False باشد یعنی کاربر تاکنون هیچ عکسی را انتخاب نکرده است. در نتیجه عکس‌های پیش فرضی که برای کاربران زن و مرد در نظر گرفته شده است بنا به جنسیت کاربر روی صفحه به نمایش در می‌آید. اما چنانچه مقدار این فیلد True باشد یعنی کاربر در سمت سرور در فایل user Image دارای تصویر می‌باشد، در نتیجه تصویر با نام userID+PassImage را از سمت سرور پیدا کرده و آن را در صفحه نمایش می‌دهد.

۳-۲-۱-۶- صفحه Chatroom:

پس از توضیحاتی که در مورد صفحه پروفایل داده شد و به اهمیت صفحه پروفایل اشاره نمودیم حال می‌خواهیم به بررسی صفحه Chatroom که یکی دیگر از صفحات مهم این پروژه است بپردازیم.

امکانات صفحه Chatroom:

- ۱) نمایش لیست تمام کسانی که در صفحه چت روم وجود دارند.
- ۲) تمام کسانی که در این صفحه وجود دارند می‌توانند با ارسال پیام^۱ با یکدیگر صحبت کنند. برای این منظور کاربران متن مورد نظر را در جعبه متن^۲ موجود نوشته سپس با کلیک کردن بر روی دکمه ارسال^۳ متن را برای دیگر کاربران ارسال می‌کنند. متن ارسال شده در پنل خاصی نمایش داده می‌شود.
- ۳) کاربران می‌توانند علاوه بر متن تصویر نما^۴ نیز ارسال کنند. به این منظور کاربر می‌بایست از جدول راهنما تصویر نماها، Tooltip مربوط به تصویر نما مورد نظر خود را انتخاب کرده و سپس این Tooltip را در جعبه متن وارد کند و دکمه ارسال را کلیک می‌کند. با این کار متن موجود در جعبه متن با نشانه مورد نظر تعویض می‌شود و عکس نشانه در پنل نمایش داده می‌شود.
- ۴) فیلتر کردن کلمات نامناسب: در صورتی که یکی از کاربران کلمه‌ای نامناسب را در جعبه متن وارد کند آن را با علامت * جایگزین می‌کند.

بررسی توابع صفحه Chatroom.aspx.cs:

تابع Page_load():

¹ Message

² TextBox

³ Send

⁴ Icon

این تابع ابتدا چک می کند که آیا این صفحه توسط کاربری تقاضا شده است یا خیر. در صورت برقرار بودن شرط نام کاربر تقاضا کننده را در متغیری به نام `uname` ذخیره می کند سپس تمام کاربران موجود در جدول `userInchatroom` در صفحه `Chatroom.aspx` و در قسمت `user online` نمایش می دهد.

تابع `webchat_Chater()`:

وظیفه این تابع فیلتر کردن کلمات نامناسبی است که کاربران در صفحه چت وارد می کنند. علاوه بر این وظیفه ارسال نشانه در صفحه چت را نیز دارد.

۳-۲-۲- معرفی بانک ها:

در این پروژه با استفاده از `Sql Server 2005` یک بانک اطلاعاتی با نام `WebChat` طراحی شده است. این بانک شامل جداول زیر می باشد.

۱- جدول `Register`: تمامی اطلاعات اعضا سایت در این جدول نگهداری می شود.

فیلدهای این جدول به شرح زیر هستند:

❖ نام فیلد: `FirstName`

نوع فیلد: `Varchar(50)`

توضیح فیلد: نام کاربر سایت در این فیلد قرار می گیرد و این فیلد نمی تواند تهی باشد.

❖ نام فیلد: `LastName`

نوع فیلد: `Varchar(50)`

توضیح فیلد: نام خانوادگی کاربر سایت در این فیلد قرار می گیرد و این فیلد نمی تواند تهی باشد.

❖ نام فیلد: `Gender`

نوع فیلد: `Bit`

توضیح فیلد: این فیلد نشاندهنده جنسیت کاربر است. اگر مقدار این فیلد `False` باشد کاربر مرد می باشد و چنانچه

مقدارش `True` باشد، کاربر خانم است. و این فیلد نمی تواند تهی باشد.

❖ نام فیلد: Mount

نوع فیلد: Varchar(50)

توضیح فیلد: این فیلد حاوی ماه تولد کاربر است. این فیلد می تواند تهی باشد.

❖ نام فیلد: Day

نوع فیلد: int

توضیح فیلد: این فیلد حاوی روز تولد کاربر می باشد. این فیلد می تواند تهی باشد.

❖ نام فیلد: Yera

نوع فیلد: int

توضیح فیلد: این فیلد حاوی سال تولد کاربر است. این فیلد می تواند تهی باشد.

❖ نام فیلد: Username

نوع فیلد: Varchar(50)

توضیح فیلد: نام کاربری عضو در این فیلد قرار می گیرد. این فیلد کلید اصلی جدول است و مقدار آن نمی تواند تهی باشد.

❖ نام فیلد: Password

نوع فیلد: Varchar(50)

توضیح فیلد: رمز عبور کاربر در این فیلد قرار می گیرد. و این فیلد نمی تواند تهی باشد.

❖ نام فیلد: SImage

نوع فیلد: Bit

توضیح فیلد: این فیلد وضعیت عکس کاربر را مشخص می کند. چنانچه مقدار این فیلد False باشد نشان دهنده این است که کاربر هیچ عکسی را برای پروفایل خود در نظر نگرفته است. و اگر مقدار این فیلد True باشد یعنی کاربر عکسی را برای پروفایل خود انتخاب کرده است و این عکس با نام `UserId + ImageType` در سمت سرور و در پوشه ای به نام `UserImage` ذخیره شده است.

❖ نام فیلد: LastUpdate

نوع فیلد: Varchar(50)

توضیح فیلد: این فیلد تاریخ آخرین زمانی را که کاربر پروفایل خود را ویرایش کرده است نشان می دهد و این فیلد

نمی تواند تهی باشد.

❖ نام فیلد: Country

نوع فیلد: Varchar(50)

توضیح فیلد: نام کشور کاربر در این فیلد قرار می گیرد. و این فیلد می تواند تهی باشد.

❖ نام فیلد: City

نوع فیلد: Varchar(50)

توضیح فیلد: نام شهری را که کاربر در آن سکونت دارد نشان می دهد. و این فیلد می تواند تهی باشد.

❖ نام فیلد: UserId

نوع فیلد: Bigint

توضیح فیلد: به هر کاربر یک شماره منحصر به فرد اختصاص داده می شود. این شماره ها به صورت اتوماتیک توسط خود

سرور تولید می شود. مقدار این فیلد هرگز تهی نیست.

❖ نام فیلد: ImageType

نوع فیلد: Char(10)

توضیح فیلد: هنگامی که کاربر عکسی را برای پروفایل خود انتخاب می کند پسوند آن در این فیلد ذخیره می شود. مقدار

این فیلد زمانی که کاربر هیچ عکسی را برای پروفایل خود انتخاب نکرده است تهی می باشد.

۲- جدول **UserInRoom**: هنگامی که کاربر وارد صفحه ChatRoom می شود اطلاعات آن وارد این جدول می شود و پس

از خروج آن از این صفحه اطلاعات آن پاک می شود.

❖ نام فیلد: UserName

نوع فیلد: Varchar(50)

توضیح فیلد: نام کاربر که می خواهد وارد صفحه چت شود در این فیلد قرار می گیرد. و این فیلد نمی تواند تهی باشد.

❖ نام فیلد: TimeEnter

نوع فیلد: DateTime

توضیح فیلد: شامل زمان ورود کاربر به صفحه چت می باشد و این فیلد نمی تواند تهی باشد.

❖ نام فیلد: PassFile

نوع فیلد: char(10)

توضیح فیلد: اگر کاربر در صفحه چت فایلی را ارسال کند پسوند فایل در این فیلد ذخیره می شود. و این فیلد می تواند

تهی باشد.

❖ نام فیلد: IdUserInRoom

نوع فیلد: Varchar(50)

توضیح فیلد: برای هر کاربری که وارد صفحه چت می شود یک شماره یکتا اختصاص داده می شود. و این فیلد نمی تواند

تهی باشد.

۳- جدول Mail: هنگامی که کاربر نامه ای را برای دوست خود ارسال می کند تمامی اطلاعات نامه در این جدول ثبت

می شود.

❖ نام فیلد: SenderMail

نوع فیلد: Varchar(50)

توضیح فیلد: حاوی نام فرستنده نامه است. و این فیلد نمی تواند تهی باشد.

❖ نام فیلد: ResiverMail

نوع فیلد: Varchar(50)

توضیح فیلد: نام گیرنده نامه در این فیلد قرار خواهد گرفت. و این فیلد نمی تواند تهی باشد.

❖ نام فیلد: TextMail

نوع فیلد: Varchar(1000)

توضیح فیلد: شامل متن نامه است. و این فیلد نمی تواند تهی باشد.

❖ نام فیلد: Attachfile

نوع فیلد: Varchar(250)

توضیح فیلد: نام فایل Attachment در این فیلد قرار می‌گیرد و این فیلد می‌تواند تهی باشد.

❖ نام فیلد: DateSend

نوع فیلد: Varchar(50)

توضیح فیلد: حاوی تاریخ ارسال نامه است. و این فیلد نمی‌تواند تهی باشد.

❖ نام فیلد: IdMail

نوع فیلد: BigInt

توضیح فیلد: در این فیلد به هر نامه یک شماره منحصر به فرد اختصاص داده می‌شود. و این فیلد نمی‌تواند تهی باشد.

❖ نام فیلد: Subject

نوع فیلد: BigInt

توضیح: عنوان نامه در این فیلد قرار می‌گیرد. و این فیلد می‌تواند تهی باشد.

۴- جدول File: هنگامی که کاربر فایلی را برای دوست خود ارسال می‌کند تمامی اطلاعات فایل در این جدول قرار

می‌گیرد.

❖ نام فیلد: SenderFile

نوع فیلد: Varchar(50)

توضیح فیلد: نام ارسال کننده فایل در این فیلد قرار می‌گیرد. و این فیلد نمی‌تواند تهی باشد.

❖ نام فیلد: ResiverFile

نوع فیلد: Varchar(50)

توضیح فیلد: نام دریافت کننده فایل در این فیلد قرار می‌گیرد. و این فیلد نمی‌تواند تهی باشد.

❖ نام فیلد: FileName

نوع فیلد: Varchar(50)

توضیح فیلد: نام پسوند فایل در این فیلد قرار می‌گیرد و این فیلد نمی‌تواند تهی باشد.

❖ نام فیلد: IdFile

نوع فیلد: Biugint

توضیح فیلد: به هر فایل ارسالی یک شماره منحصر به فرد اختصاص داده می شود. مقدار این فیلد نمی تواند تهی باشد

نام فیلد: FileSize

نوع فیلد: int

توضیح فیلد: سایز فایل ارسالی در این فیلد قرار می گیرد.

❖ نام فیلد: FileTimeSend

نوع فیلد: DateTime

توضیح فیلد: زمان ارسال فایل در این فیلد قرار می گیرد. این فیلد نمی تواند تهی باشد.

۵- جدول Invitation: هنگامی که کاربر دعوتنامه ای را برای کاربر دیگری ارسال می کند تمامی اطلاعات فرم دعوتنامه در

این جدول ثبت می شود.

❖ نام فیلد: SenderInvitation

نوع فیلد: Varchar(50)

توضیح فیلد: شامل نام ارسال کننده نامه است. و این فیلد نمی تواند تهی باشد.

❖ نام فیلد: ResiverInvitation

نوع فیلد: Varchar(50)

توضیح فیلد: شامل نام دریافت کننده نامه است. و این فیلد نمی تواند تهی باشد.

❖ نام فیلد: InvitationText

نوع فیلد: Varchar(1000)

توضیح فیلد: متن دعوتنامه در این فیلد قرار می گیرد و این فیلد نمی تواند تهی باشد.

❖ نام فیلد: SInvitation

نوع فیلد: bit

توضیح فیلد: وضعیت پاسخگویی به دعوت نامه را مشخص می کند. مقدار پیش فرض این فیلد False است و این فیلد

نمی تواند تهی باشد.

❖ نام فیلد: IdInvitation

نوع فیلد: bigint

توضیح فیلد: به هر دعوت نامه یک عدد منحصر به فرد اختصاص می‌دهیم. و این فیلد نمی‌تواند تهی باشد.

❖ نام فیلد: InvitationDate

نوع فیلد: Varchar(50)

توضیح فیلد: تاریخ ارسال دعوتنامه در این فیلد قرار می‌گیرد. و این فیلد نمی‌تواند تهی باشد.

۶- جدول Friend: اسامی تمامی دوستان یک کاربر در این جدول ثبت می‌شود.

❖ نام فیلد: FirstUser

نوع فیلد: Varchar(50)

توضیح فیلد: نام کاربر X در این فیلد قرار می‌گیرد. و این فیلد نمی‌تواند تهی باشد.

❖ نام فیلد: SeconderUser

نوع فیلد: Varchar(50)

توضیح فیلد: نام دوست کاربر X در این فیلد قرار می‌گیرد. و این فیلد نمی‌تواند تهی باشد.

❖ نام فیلد: Statuse

نوع فیلد: Varchar(50)

توضیح فیلد: وضعیت دوست را نشان می‌دهد. اگر مقدار این فیلد False باشد یعنی SeconderUser در شبکه وجود

ندارد و یا وجود دارد اما به صورت Invisiable است و اگر مقدار آن True باشد یعنی SeconderUser بر روی

شبکه موجود است. و این فیلد نمی‌تواند تهی باشد.

فصل چهارم: نتیجه گیری

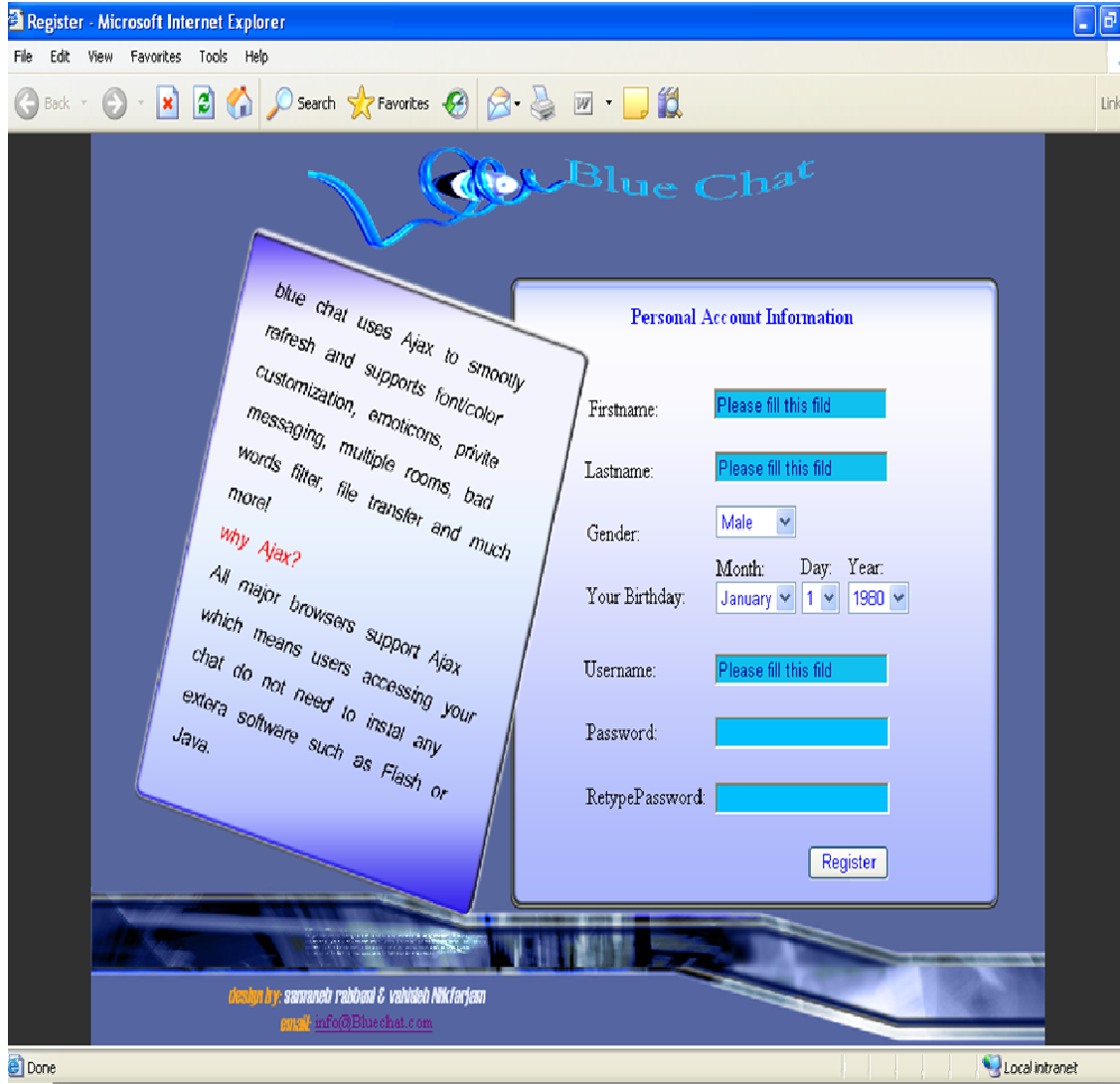
نتیجه گیری:

نرم افزاری که در این پروژه ارائه شد نمونه ای از یک نرم افزار وب چت تحت وب بود. در این پروژه سعی کردیم نرم افزاری را ارائه دهیم که با فناوری های جدید مانند آجاکس طراحی شده است و دارای قابلیت های قابل ملاحظه ای برای کاربران است. این نرم افزار همانند دیگر نرم افزارها دارای قابلیت ها و معایب خاص خود است که قابلیت ها و مزایای آن در فصل ها قبل ذکر شده است.

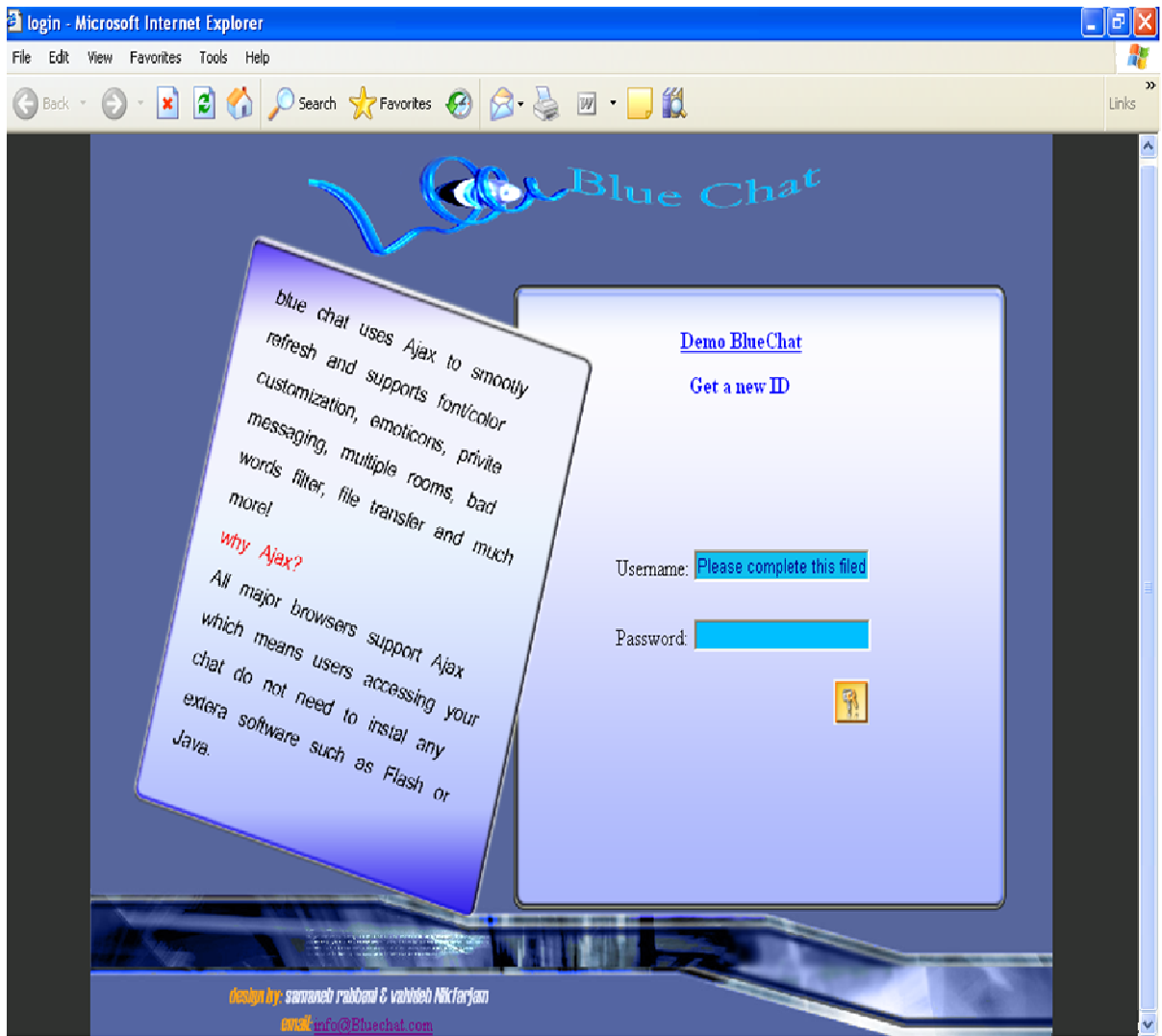
در این پروژه با توجه به زمان محدودی که داشتیم سعی نمودیم حتی الامکان قابلیت های بیشتری نسبت به آنچه در عنوان پروژه و توسط استاد محترم جناب آقای مهندس خادم زاده مطرح شده بود ارائه دهیم. اما این نرم افزار همانند سایر نرم افزارها می توانند گسترش یابد و سایر نیازهای کاربران را بر طرف نماید. البته ناگفته نماند بسط و گسترش این نرم افزار وقت و تحقیق بیشتری را می طلبد.

پیوست ۱

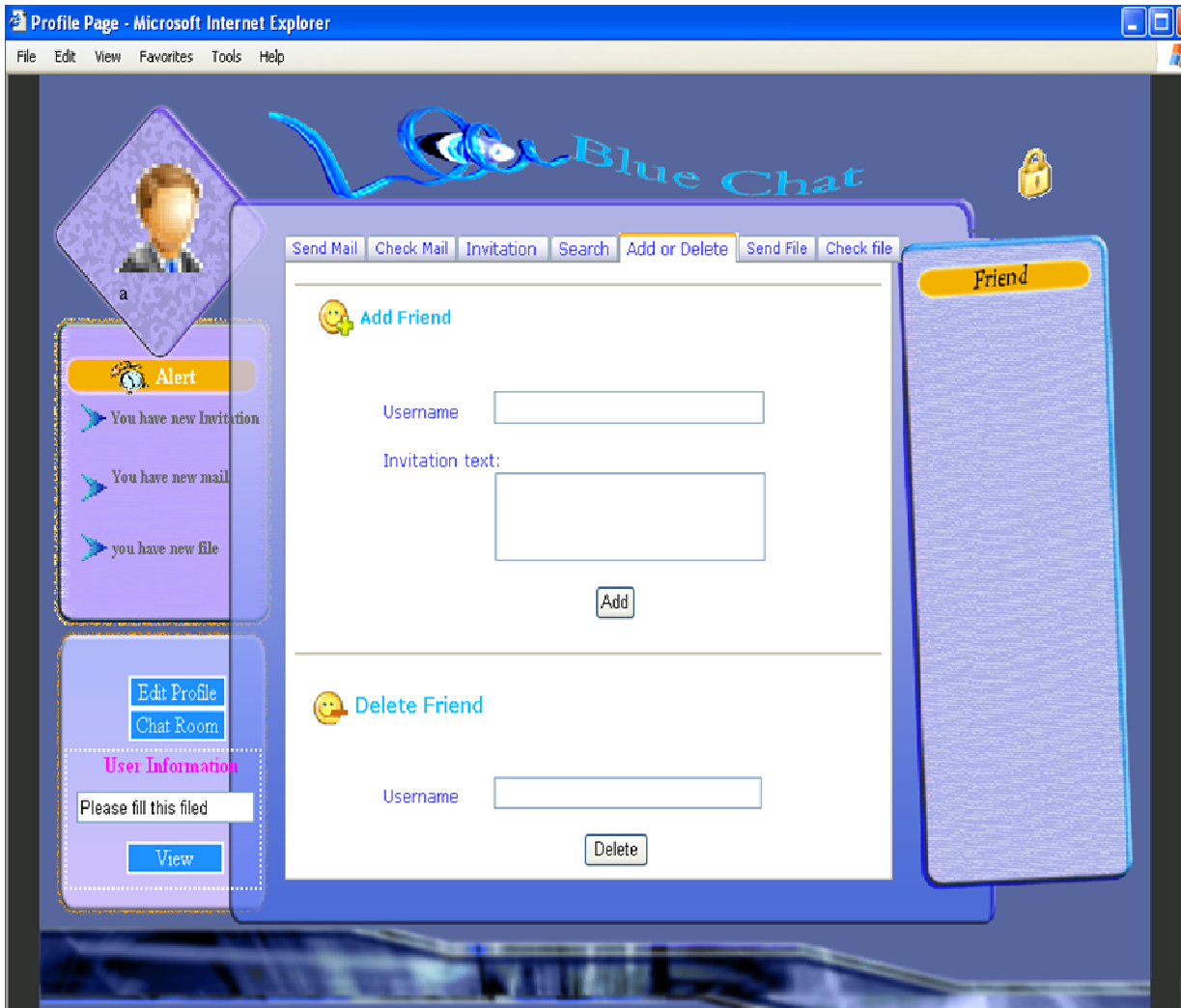
صفحة Registerarion



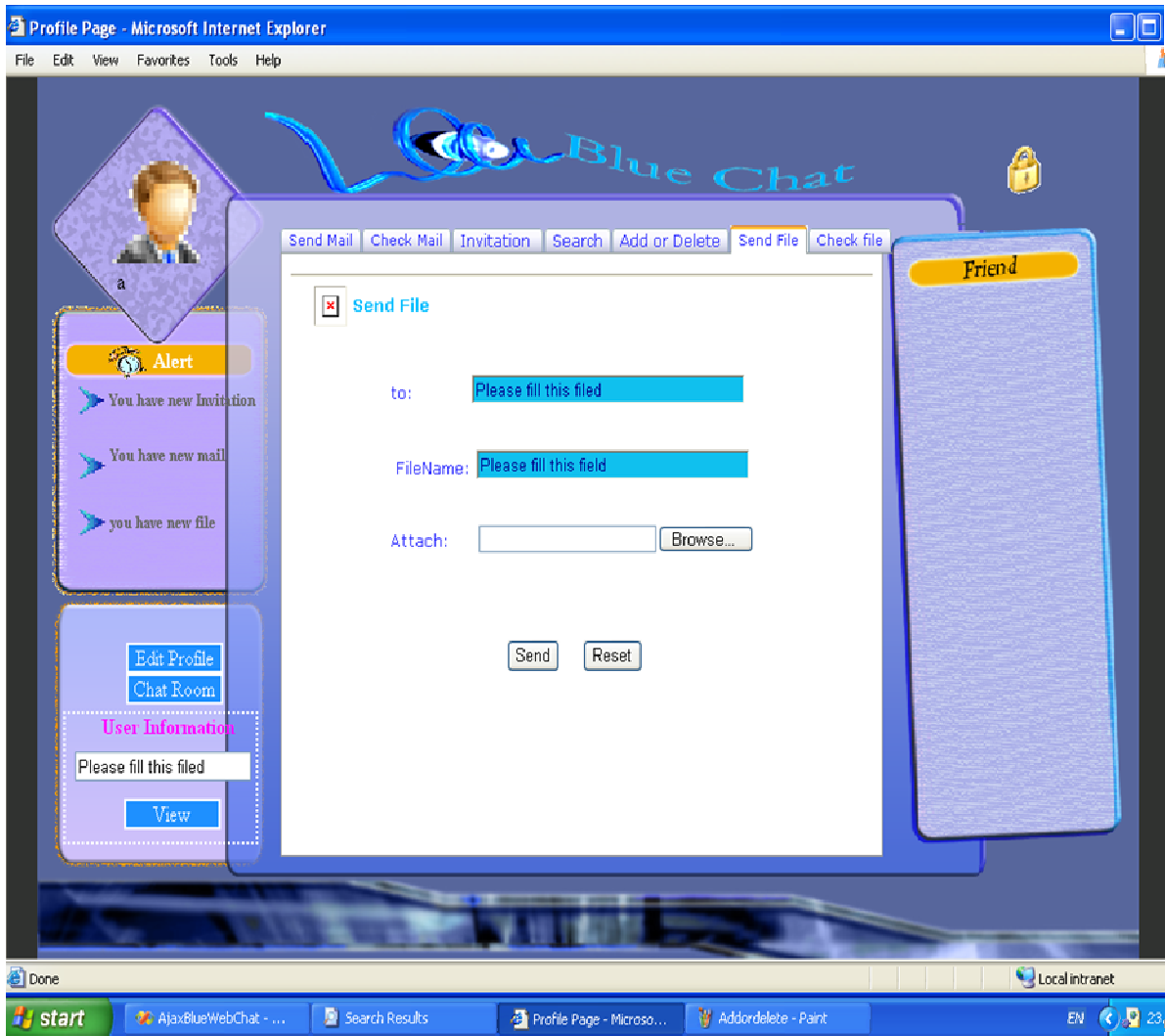
صفحه Login



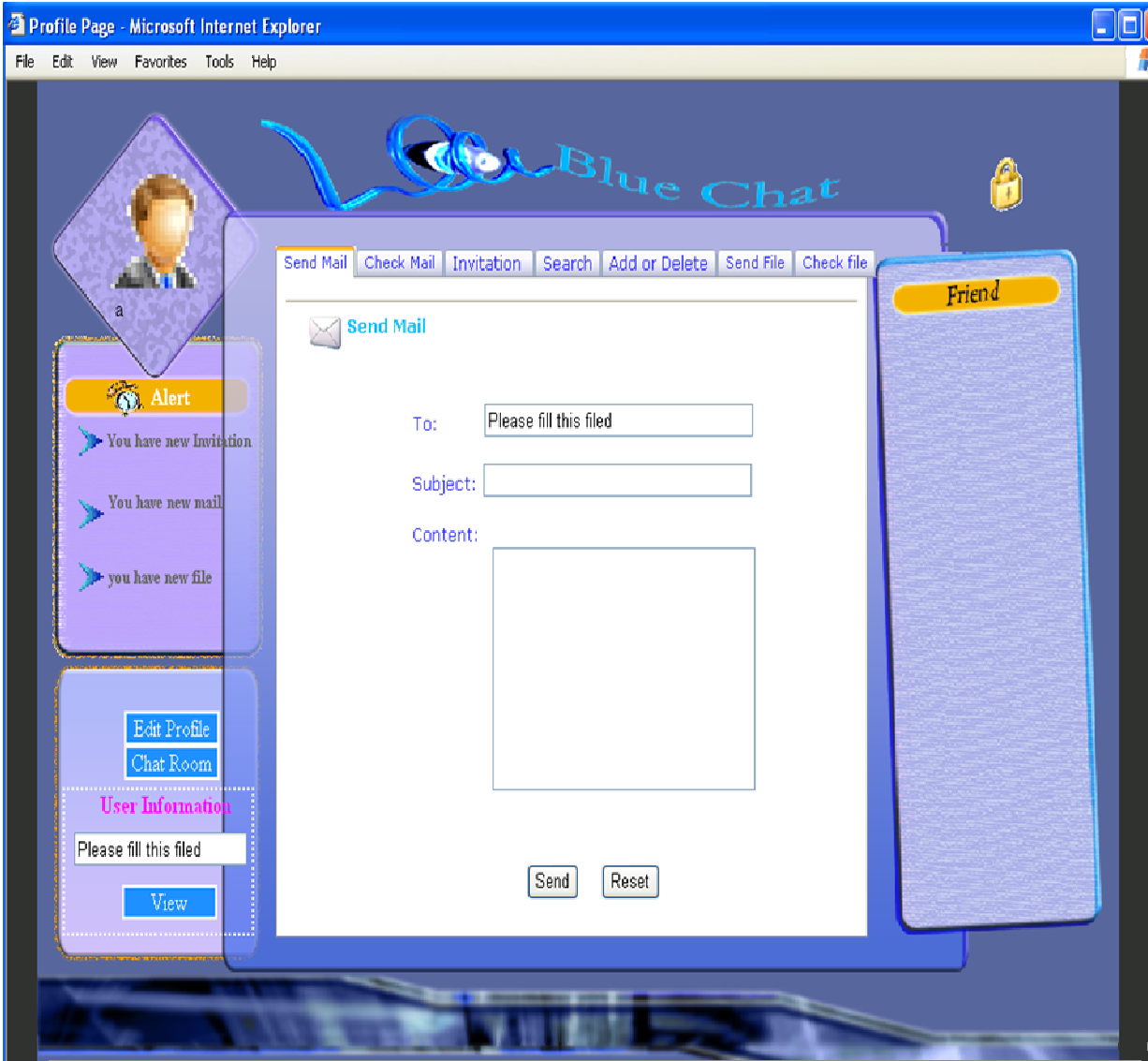
صفحه profile (اضافه و کم کردن دوست)



صفحه Profile (ارسال فایل)



صفحه Profile (ارسال نامه الکتونیکی)



صفحه EditProfile

Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Refresh Print Copy Paste

Address: http://localhost:1043/AjaxBlueWebChat/EditProfile.aspx?Username=s

Blue Chat

Edit Profile

Personal Information Password

FirstName:

LastName:

Country:

City:

Gender: Male

Birthday: month: day: Year:

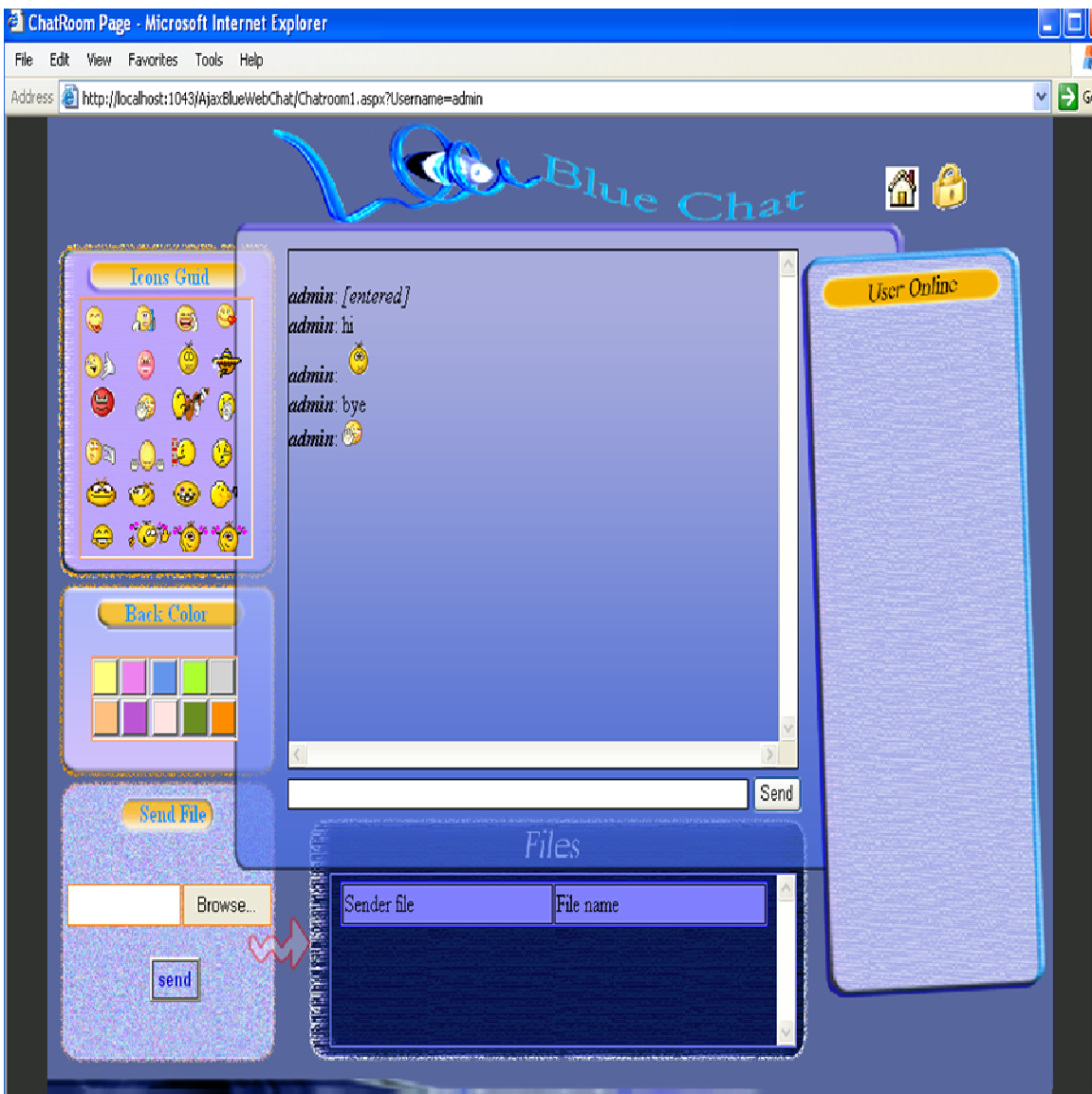
Design by: Samaneh Rabbani & Vahideh Nikfarjam
Email:

صفحه UserInformation

The screenshot shows a Microsoft Internet Explorer browser window displaying a web page titled "UserInformation Page". The address bar shows the URL: `http://localhost:1043/AjaxBlueWebChat/UserInformation.aspx?Username=a`. The page features a blue-themed design with a "Blue Chat" logo at the top. A central purple box contains the "User Information" for a user named "alunad". To the left of the information box is a diamond-shaped placeholder for a user profile picture. At the bottom of the page, there is a credit line: "Design by: Samaneh Rabbani & Vahideh Nikfarjam".

User Information	
FirstName:	alunad
LastName:	khademzadeh
Country:	Iran
City:	mashad
Gender:	Male
Birthday:	1/1/1980
Last Update:	12-09-2008 19:34:32

صفحة Chatroom



مراجع و منابع

مراجع و منابع:

۱- دوتی جی، اندرو، هراتیان، سعید و فلاح، مهدی، آموزش گام به گام ASP.NET، موسسه فرهنگی هنری نقش سیمرغ، ۱۳۸۱

۲- والتر، استفان، احترامی، بابک، انتشارات دانش نگار دانشگاه امیر کبیر، ۱۳۸۲

۳- وی مایر، ریچارد و ساوتل، ریک، گوهری، پریا، sql server 2000، موسسه علمی فرهنگی نص، ۱۳۸۲

4- <http://www.microsoft.com/SQL/>

5- <http://www.asp.net/Ajax>

6- <http://www.codeproject.com/>