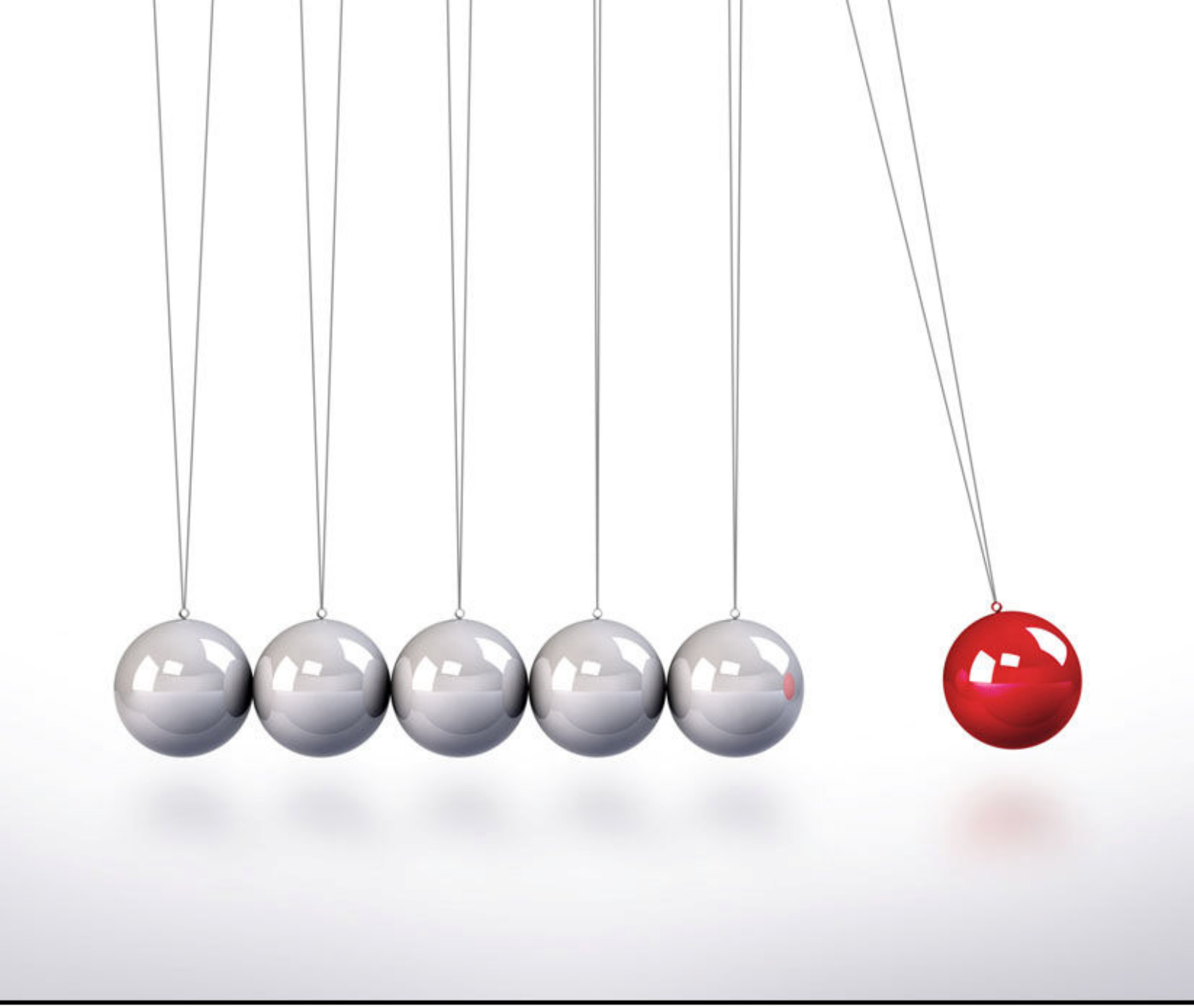


AJAX

راه حلی نوین برای برنامه های تحت وب

استاد مربوطه: مهندس حسن بشیری

سجاد رضایی
دانشگاه صنعتی همدان
sajjad.rezaee@yahoo.com

به نام خالق هستی بخش

AJAX

راه حلی نوین برای برنامه‌های تحت وب

به عنوان ارائهٔ کتبی

در درس

شیوه ارائه مطالب علمی و فنی

گروه کامپیوتر

دانشکده برق و کامپیوتر

دانشگاه صنعتی همدان

استاد راهنما: مهندس حسن بشیری

دانشجو: سجاد رضایی

شهریور ۱۳۸۸

sajjad.rezaee@yahoo.com

فهرست مطالب

فهرست شکل‌ها و جدول‌ها

فصل ۱

مقدمه‌ای بر برنامه‌های تحت وب

۱-۱: تاریخچه مختصری از وب

در اوایل دهه نود، تور جهان گستر یا وب در حالی متولد شد که شبکه اینترنت حدود بیست سال سن داشت. به جرأت می‌توان گفت تا قبل از تولد وب، فقط دانشگاهیان و پژوهشگران در خصوص اینترنت چیزی می دانستند یا از خدمات بهره می بردند. تا قبل از ظهور وب عمومی‌ترین کاربرد اینترنت، سیستم پست الکترونیکی بود. وب، اینترنت را به شبکه همگانی تبدیل کرد. سیطرهٔ نفوذ آن را به اعماق لایه های اجتماع متمدن امروز کشاند و در زندگی روزمره مردم جای خود را باز کرد. امروزه کاربران وب خود یک جامعه بزرگ و جهانی تشکیل داده اند، جامعه ای که همانند جوامع بشری دیگر، نیک و بد را در کنار هم دارد! جامعه وب بدون مرز و جهانی است.

در خلال پانزده سال اخیر، وب تحولات و پیشرفتهای شگرفی داشته است. زبان‌های وب بسیار قدرتمند و قابل انعطاف شده اند. سرویس دهنده های وب خدمات بسیار قدرتمند و وسیع تری را عرضه می کنند و مرورگرها روز به روز جذاب تر، مجهز به امکانات بیشتر و کارآمدتر شده اند. مجموعه این عوامل باعث شده که وب از شکل یک رسانه ارتباط جمعی خارج شده و در کلیه شئونات زندگی بشر وارد شود. امروزه صحبت از تجارت

الکترونیکی، دولت الکترونیکی، آموزش الکترونیکی و بسیاری از چیزهای الکترونیکی دیگر بر سر هر زبانی شنیده می شود.

۱-۲: مفهوم وب

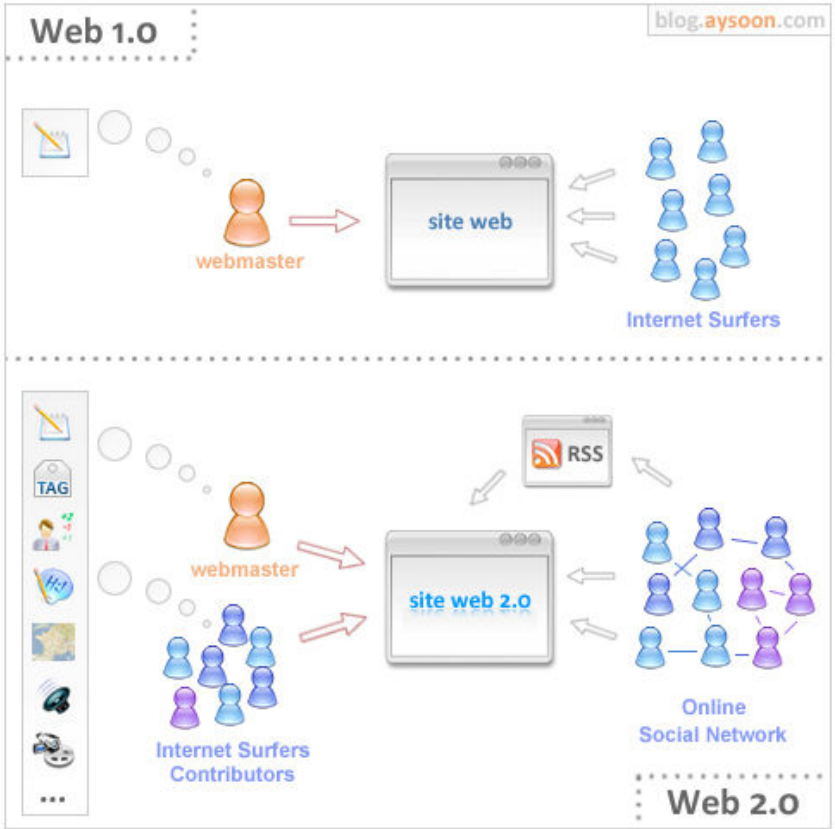
تور جهان گستر یا «وب»، یک روش معماری یا به عبارتی یک نظام برای ذخیره سازی، سازماندهی و دسترسی به مستندات به هم پیوند خورده‌ای است که بر روی هزاران ماشین در کل جهان پراکنده و توزیع شده‌اند. هر یک از این مستندات پیوند خورده حاوی متن، صدا، تصویر، کد اجرایی، اسکریپت و نظائر آن هستند و می توانند به سند یا اسناد دیگر واقع بر ماشینی متفاوت از این جهان بزرگ اشاره نمایند. بزرگترین مزیت وب سادگی استفاده از آن است؛ کاربر با وارد کردن آدرس یک وب سایت، صفحه اصلی آن را بر روی کامپیوتر خود متنقل می کند؛ آن را نگاه می کند و مطالعه نموده و اگر تمایل به دریافت اطلاعات بیشتری در خصوص آیتم‌هایی که پررنگ هستند، داشت با ماوس خود روی آن کلیک می نماید. (آیتمهای پررنگ نقطه پیوند صفحه فعلی با صفحات دیگر به حساب می آیند.) با کلیک بر روی یکی از این پیوندها، صفحه جدید هم مثل صفحه قبلی بر روی کامپیوتر او بارگذاری شده و این روند ادامه می‌یابد. کاربر عموماً متوجه نیست که هر صفحه از چه نقطه ای از جهان بر روی کامپیوترش منتقل شده است. بدیهی است که در ذیل هر آیتم پررنگ[1] یک آدرس دیگر وجود دارد که ارتباط بین دو صفحه را برقرار کرده است. این آدرس خاص که URL نام دارد موقعیت دقیق صفحات ذخیره شده در هر نقطه از جهان را مشخص می کند. URL را یک آدرس استاندارد و جهانی فرض کنید که موقعیت هر منبع[2] همانند صفحات وب، فایل‌های داده، صدا، تصویر، اسکریپت‌های اجرایی و نظائر آنها را به دقت مشخص می نماید. شاید وب جذاب‌ترین جنبه استفاده از مدل انتزاعی «سرویس دهنده/مشتری» (Client/Server) در شبکه اینترنت باشد. برنامه سمت مشتری (یعنی مرورگر) تقاضایی را برای دریافت یک صفحه وب یا یک فایل به سمت سرویس دهنده دلخواه در هر نقطه از جهان ارسال می‌دارد. برنامه سرویس دهنده با دریافت تقاضا و در صورت داشتن مجوز، آن را پذیرفته و داده ها و پاسخ مناسب را بر می گرداند.

---

[1] hyperlink
[2] resource

۱-۳: نسل اول و نسل دوم وب

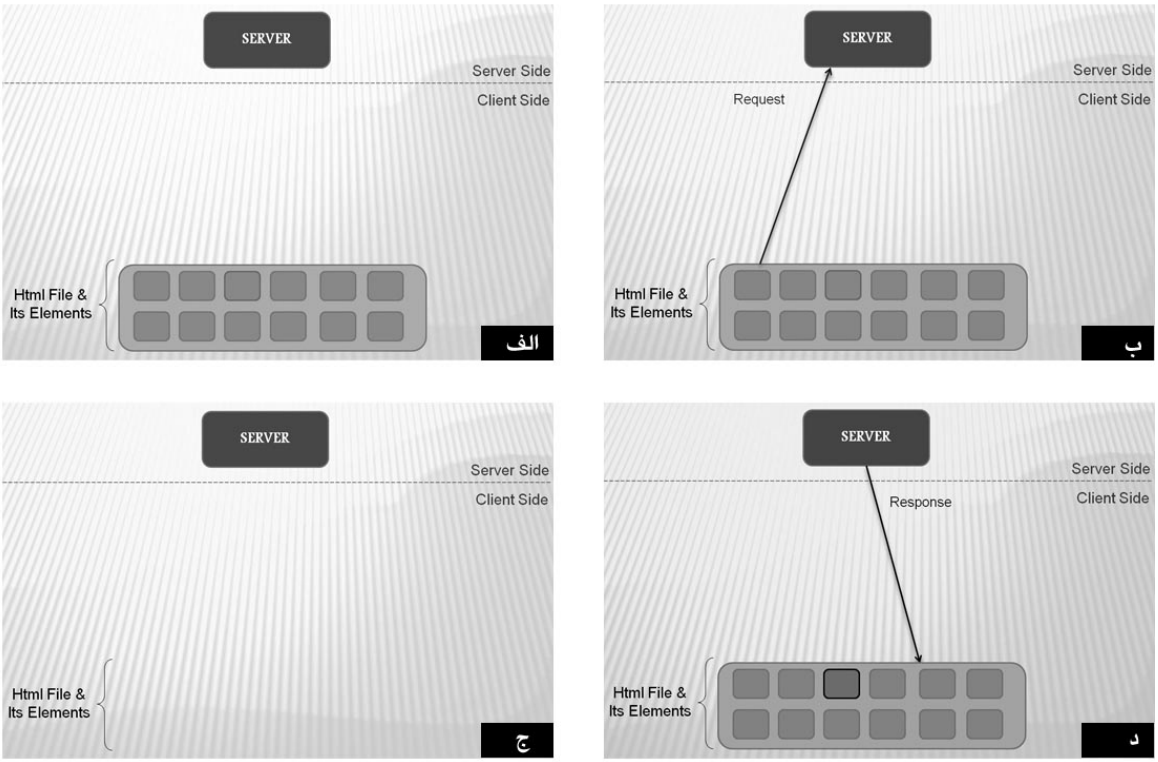
اولین نسل از وب (وب۱) تشکیل شده از صفحات ایستا (static) بود. بدین معنا که مدیریت محتوای صفحات باید به صورت دستی و برای هر صفحه به صورت جداگانه صورت می‌گرفت. همچنین مفاهیم پایگاه‌های داده و برنامه نویسی سمت سرور مفهوم چندانی نداشت و معمولا برای هر صفحه وب باید یک فایل مجزا وجود داشت. اما در نسل دوم وب (وب۲) با حضور مفاهیم پایگاه‌های داده و برنامه نویسی سمت سرور، این قابلیت به وجود آمد که صفحات وب به صورت پویا (dynamic) ساخته شوند. بدین معنا که محتوای یک صفحه لزوما یک فایل ذخیره شده در کامپیوتر سرور نیست، بلکه درخواست‌های کاربر در سمت سرور پردازش شده، سپس صفحه مورد نظر ساخته شده و برای کاربر ارسال می‌شود. از آن جایی که قابلیت ذخیره اطلاعات به صورت سازماندهی شده در پایگاه‌های داده وجود دارد، مدیر سایت می‌تواند مطالب را در پایگاه-های داده ذخیره کند و بر آن‌ها مدیریت (چون مرتب سازی، جستجو و ...)داشته باشد. همچنین در وب۲ امکان مدیریت محتوا و ساخت صفحات وب از طریق وب هم امکانپذیر است. سرویس‌هایی همچون وبلاگ‌ها، انجمن‌های اینترنتی، آلبوم تصاویر و سایر سرویس‌های مشابه در مجموعه وب۲ قرار می‌گیرند.



شکل ۱-۱: مقایسه وب۱ و وب۲

۱-۴: نحوه عملکرد وب کلاسیک(قبل از ظهور AJAX)

نحوه عملکرد وب قبل از ظهور AJAX بدین صورت بود:
پس از این که برای اولین بار صفحه مورد نظر بارگذاری شد (شکل ۱-۲-الف)، کاربر تقاضایی را از طریق مرورگر وب خود به سرور ارسال می‌کرد (شکل ۱-۲-ب)، سپس ارتباطش با نرم افزار قطع گشته و با صفحه‌ای سفید مواجه می‌گشت تا پردازش‌های سمت سرور انجام شود (شکل ۱-۲-ج). پس از ارسال پاسخ توسط سرور قسمت مورد نظر در صفحه به روز می‌شود. (شکل ۱-۲-د) در وب کلاسیک اصلا مهم نیست که درخواست شما چه مقدار تغییراتی در صفحه وب اعمال کند و در هر پاسخی باید کل صفحه مجدداً توسط سرور ارسال گردد. یعنی حتی قسمت‌های ثابت صفحه همچون سربرگ، نمایه‌ی سایت و منو هم در هر تقاضا باید دوباره بارگذاری شود. و این باعث هدر رفتن پهنای باند شبکه خواهد شد.



شکل ۱-۲: نحوه عملکرد وب کلاسیک

۱-۵: مقایسه برنامه های تحت وب[1] با برنامه های رومیزی[2]

اگر بخواهیم برنامه‌های تحت وب را با برنامه‌های تحت وب(قبل از ظهور AJAX) مقایسه کنیم، به جدول زیر می‌رسیم:

جدول ۱-۱: مقایسه برنامه های تحت وب با برنامه های رومیزی

برنامه‌های رومیزی	برنامه های تحت وب
سرعت اجرای بالا	سرعت پایین تر
ارتباط پیوسته کاربر در حین اجرای برنامه	وجود refresh در حین کار با برنامه
امکانات گرافیکی زیاد و محیط پویا	امکانات گرافیگی کمتر
عدم به روز بودن	به روز بودن
عدم اجرا در همه سیستم عامل ها	اجرا در تمامی سیستم های عامل

البته تفاوت‌های دیگری بین این دو نوع از برنامه‌ها وجود دارد، که از جمله می‌توان به امنیت بیشتر نرم افزارهای رومیزی اشاره کرد. از آن‌جا شما خود مدیریت نرم افزارهای رومیزی را بر عهده دارید نرم افزارهای رومیزی خطرات امنیتی کمتری نسبت به نرم افزارهای تحت وب دارند. زیرا نرم افزارهای تحت وب روی شبکه‌ای بسیار بزرگ از کامپیوترها اجرا می‌شوند و میلیون‌ها کاربر به آن دسترسی دارند.

[1] Web Applications
[2] Desktop Applications

فصل ۲

AJAX چیست؟

۲-۱: تعریف AJAX

AJAX، مخفف عبارت Asynchronou Javascript And XML است به معنی استفاده از جاوا اسکریپت و XML در تراکنشهای همگام. اگر بخواهیم در معنی لغات دقت کنیم، باید بگوییم که AJAX تکنولوژی جدیدی نیست، بلکه فقط تکنیک‌های شناخته شده برنامه نویسی وب را به گونه‌ای غیرمعمول ترکیب می‌نماید تا به برنامه نویسان وب این امکان را بدهد که برنامه‌های وب را به صورت جذابتر از آن چیزی که به صورت معمول رواج دارد، تولید کنند. وقتی ما با برنامه‌های کاربردی که تحت وب نیستند (برنامه‌های رومیزی) کار می کنیم، انتظار داریم نتایج عملیاتی که انجام می دهیم، فوراً تولید و نمایش داده شوند، بدون نیاز به اینکه ما منتظر باشیم تا کل صفحه نمایش برنامه دوباره توسط برنامه تولید شود. مثلاً هنگامی که ما با نرم افزاری مثل اکسل کار می کنیم، انتظار داریم اثر تغییراتی که در یک ستون ایجاد می کنیم وقتی که مشغول ورود اطلاعات دیگر یا کار با ماوس هستیم، بی درنگ به ستونها و سلولهای مرتبط منتقل شود.

متاسفانه اینگونه تعامل برای کاربران برنامه های تحت وب بسیار کم در دسترس بوده است. بسیاری این تجربه را داشته اند که فیلدهای اطلاعاتی یک فرم وب را پر کرده اند و روی یک دکمه یا لینک کلیک نموده و منتظر مانده اند تا صفحه وب به آرامی تغییر کند و

نتایج به دست‌آمده را نمایش دهد. به علاوه، بسیار شاهد بوده‌ایم که صفحات جدید شامل مولفه‌های یکسانی با صفحه قبل می باشد و نیازی به بارگذاری مجدد نداشته‌اند. عکس-های زمینه، لوگوها و منوها از این دسته مولفه‌ها هستند. AJAX، نوید حل این مشکل را به ما می‌دهد. AJAX، به صورت یک لایه اضافی بین مرورگرِ کاربر و سرورِ وب عمل نموده و ارتباطات سرور را در پس‌زمینه[1] پیاده سازی می کند، درخواست ها را به سرور ارسال و نتایج آن را پردازش می نماید. نتایج به دست آمده ممکن است با محتویات صفحه در حال نمایش ترکیب شده و به نمایش درآیند، بدون اینکه نیاز به بارگذاری مجدد تمام صفحه وب باشد.

در برنامه های AJAX حتماً نیاز نیست که درخواست ها حساس به عمل کاربر مثل کلیکِ دکمه یا لینک باشد. در یک برنامهٔ AJAX با نگارش خوب، ممکن است قبل از اینکه کاربر انجام عملی را بخواهد، درخواست مورد نظر به سرور ارسال و نتایج آن دریافت و به کاربر نشان داده شود. این معنی کلمه غیر همگام (asynchronous) در عبارتی که AJAX مخفف آن است، می باشد.

قسمتی از برنامه AJAX که در پس‌زمینه مرورگر کاربر انجام می شود –مثل ارسال درخواست به سرور و پردازش نتایج آن– توسط جاوا اسکریپت انجام می شود وXML به عنوان ابزار رایج برای کدینگ و فرمت ارسال اطلاعات توسط AJAX استفاده می شود، تا اطلاعات بین کلاینت و سرور به صورت کارآتر منتقل شوند.

۲-۲: فناوری‌های مورد استفاده در AJAX

گفته شد که AJAX یک فناوری نیست، بلکه به کارگیری چند فناوری موجود در مسیری متفاوت است. این فناوری‌ها عبارتند از:

- زبان HTML و CSS
- جاوا اسکریپت
- زبان XML
- مدل شیء گرای سند (DOM)
- شیء XmlHttpRequest

۲-۲-۱: زبان HTML و CSS

[1] background

HTML، مخفف کلمه hyper text markup language می‌باشد و عموماً فایلهای این کدنویسی با پسوند html و htm معرفی می‌شوند.
هر فایل HTML حاوی برچسب‌های (تگ‌ها) کوچک نشانه گذاریست و این برچسب‌ها هستند که به مرورگر بیان می‌کنند که صفحه را چگونه نمایش دهد و یک فایل HTML می‌تواند توسط یک ویرایشگر متنی ساده همچون notepad ساخته شود.
کدنویسی در این زبان بسیار ساده است و نیاز به صرف زمان اندکی دارد و می‌توانید در مدت یک هفته به این زبان مسلط شوید.
به عنوان مثال کد زیر را تحلیل می‌کنیم :

```
<html>
<head>
        <title/>عنوان صفحه<title>
     <head/>
<body>
       متن معمولی<b> متن درشت خط </b>
     <body/>
<html/>
```

اگر دقت کنید هرکدی که آغاز می‌شود در خطهای بعدی به کد خاتمه خود ختم می‌شود. بطور مثال: برچسب <html> که با </html> به پایان می‌رسد.
در واقع اولین برچسب در فایل HTML شما، <html> است.این برچسب به مرورگر شما می‌فهماند که این آغاز یک فایل HTML است، آخرین برچسب در فایل شما </html> است.این برچسب به فایل شما می‌فهماند که این انتهای یک فایل HTML است.
مطالب واقع در میان برچسب head در نمای کلی صفحه نمایش داده نمی‌شوند و کاربرد خارج از قسمت نمایش اصلی (body) هستند. همانگونه که مشاهده می‌کنید برچسب title در دل این برچسب قرار گرفته است و وظیفه نمایش عنوان صفحه را دارد.
محتویات سایت شما در بین برچسب body قرار می‌گیرد و متنی که در میان <b> قرار می‌گیرد به شکل درشت خط و یا Bold نمایش داده می‌شود. توجه : در کدنویسی به این سبک B و b تفاوتی ندارند ولی بهتر است خود را با حروف کوچک عادت دهید.
حال زمان آنست که با برخی از برچسب‌ها و عناصر دیگر html آشنا شوید:
عنصرBgcolor به منظور معرفی یک زمینه برای صفحه یا جدول شما کاربرد دارد و در body به این شکل اضافه می‌شود : <body bgcolor="green"> و در بین برچسب body قرار می‌گیرد. توجه: عناصر همیشه به برچسب شروع عناصر HTML اضافه می‌شوند.

برچسب table به منظور ایجاد یک جدول در صفحه می‌باشد.
عنصر border برای تعیین اندازه لبه‌های جدول کاربرد دارد و در میان برچسب جدول مورد نظر قرار می‌گیرد. برای قرار گرفتن در بین جدول مورد نظر باید به شکل <table border="0"> وارد کنید. توجه: عناصر همیشه در یک زوج نام/مقدار می-آیند، مانند : نام="مقدار"

۲-۲-۲: جاوا اسکریپت

HTML، که زبان ساخت صفحات وب است امکانات محدودی دارد. اینترنت در اصل برای انتقال داده‌ها و رساله‌های علمی طراحی شد. در آن زمانی زبانی مورد نیاز بود که بتواند متن را فرمت کند و تصاویر را به نمایش در آورد. طرح‌های تخیلی و جلوه‌های چندرسانه‌ای لازم نبود، در نتیجه، موقع طراحی HTML این امکانات در نظر گرفته نشد. با محبوب‌شدن فوق‌العاده‌ی اینترنت، مشخص شد که این زبان قدرت کافی برای طراحان وب ندارد، و در نتیجه زبان‌هایی چون جاوا اسکریپت (JavaScript) به میدان آمدند.

۲-۲-۲-۱: چرا جاوا اسکریپت؟

جاوا اسکریپت یک زبان برنامه‌سازی ساده با فرمان‌هایی است که می‌تواند مستقیما در HTML جای بگیرد و به همراه HTML به وسیله‌ی یک مرورگر تفسیر شود. بدین معنی که کاربران مجبور نخواهند بود برنامه‌ها و یا فایل‌های ویژه‌ای را برای تماشای صفحات حاوی جاوا اسکریپت دریافت و بر روی سیستم خود نصب کنند، زیرا برنامه‌ی مرورگر آن‌ها می‌تواند کد نهاده‌شده‌ی جاوا اسکریپت را به همان ترتیبی بخواند که HTML را می‌خواند.
جاوا اسکریپت حاوی اطلاعاتی است که به برنامه‌ی مرورگر می‌گوید که به دنبال کدام ورودی بگردد و با آن چه کند. این زبان برای ردیابی هر نوع ورودی‌ای، مانند داده‌هایی که به وسیله‌ی حرکت نشانگر ماوس حاصل می‌شود، طراحی شد. پس از دریافت ورودی می‌تواند آن را پردازش کند و براساس محاسبات خود محتویاتی جدید به وجود بیاورد، یا جلوه‌های ویژه‌ای را به راه بیاندازد.
استانداردهای چندرسانه‌ای فراوانی وجود دارد که کارآمدتر از خروجی‌های جاوا اسکریپت است، اما محبوبیت جاوا اسکریپت به این زودی‌ها از بین نخواهد رفت، مهم‌تر از همه یادگیری ساده‌ی آن است و همچنین می‌توان از آن به عنوان یک زبان اسکریپت عالی

نام برد. این زبان به عنوان یک زبان عالی برای آموزش مبتدیان مطرح می‌شود. وقتی مبانی را یاد بگیرید بخش پیشرفته‌ی آن کمی پیچیده‌تر از HTML می‌باشد.
سادگی جاوا اسکریپت این مزیت را دارد که باعث می‌شود فایل غیرضروری به یک صفحه‌ی وب اضافه نشود و در نتیجه زمان لود صفحه به حداقل برسد.
مزیت دیگر جاوا اسکریپت آن است که اکثر جلوه‌های آن در واقع مرور وب را بهینه می‌سازد و راه بازدیدکنندگان را منحرف نمی‌کند. محاسبه‌گرهای جاوا اسکریپت می‌توانند در یک صفحه‌ی تنها جای بگیرند، برخلاف محاسبه‌گرهای معمولی که از کاربر می‌خواهند داده‌ها را وارد کند، آن را تسلیم کند و نتیجه را بر روی یک صفحه‌ی مجزای دیگر ببیند.
جاوا اسکریپت اغلب برای ساخت جلوه‌های rollover که در بسیاری از صفحات وب می‌بینید به کار می‌رود، مانند تغییر رنگ لینک یا تغییر تصویر پس از قرارگرفتن نشانگر ماوس روی آن. یکی از کاربردهای مفید جاوا اسکریپت آن است که می‌تواند برای ساخت فرم‌های دینامیکی‌ای به کار رود که براساس ورودی کاربر تغییر می‌کند، بی آن که لازم باشد که کاربر پیوسته برای بارکردن صفحات جدید در انتظار بماند.
یکی دیگر از مزیت‌های فوق‌العاده جاوااسکریپت این است که به هیچ نرم‌افزار یا پلاگین (plug-in) کمکی برای اجرا نیازمند نیست. شما برای دیدن یک فایل فلش به برنامه‌ی Flash Player، برای گوش دادن یک آهنگ به برنامه‌ی Real Player و... نیازمندید ولی برای جاوا اسکریپت هیچ‌کدام از این‌ها نیاز نیست.

۲-۲-۲-۲: امنیت جاوا اسکریپت

خوشبختانه هرروز امنیت جاوا اسکریپت بالاتر می‌رود؛ نگارش‌های جدید مرورگرها کاربران را بسان یک ژنرال در برابر حملات مخرب جاواسکریپت مقاوم می‌سازند.

۲-۲-۳: زبان XML

XML، مخفف eXtensible Markup Language و همچون Html، یک زبان نشانه گذاری است. XML، یک زبان بسیار رایج برای تبادل داده و سازماندهی داده‌ها برای به اشتراک گذاری است، چون این قابلیت را دارد که داده‌ها را رده‌بندی کرده و نقش‌های بسیاری دقیقی برای قالب‌بندی داده‌ها خلق کند. همچنین می‌توان از آن را در محل‌های مختلف خروجی گرفت. برای مثال یک سری داده خاص به این فرمت را می‌توان در پایگاه‌های داده، صفحات وب و یا یک فرم چاپی استفاده کرد. همچنین XML، به وابسته و محدود به یک دستگاه یا پلت‌فرم خاص نیست.

تگ‌های XML، از پیش تعریف شده نیست. بر خلاف HTML، که تگ‌ها مشخص و ثابت هستند، در XML، شما باید تگ‌ها را خودتان تعریف کنید.

به عنوان مثال کد XML زیر لیستی از چند هتل را تعریف می‌کند. هر هتل دارای نام، تعداد اتاق‌ها و نام شهری که در آن قرار گرفته است، می‌باشد.

```
<HotelList>
      <Hotel>
            <Name>Hotel Shakespeare</Name>
            <Rooms>50</Rooms>
            <City>Birmingham</City>
      </Hotel>
      <Hotel>
            <Name>Hotel Washington</Name>
            <Rooms>500</Rooms>
            City>Chicago</City>
      </Hotel>
</HotelList>
```

۲-۲-۴: مدل شیءگرای سند (DOM)

مدل شیءگرای سند یا DOM (Document Object Model) عنوان یکی از دو معماری عمده است که بر اساس آن سندهای اکس‌ام‌ال(از جمله HTML) را به اشیای موجود در آن، تجزیه نموده، و آن‌ها را به‌صورت یک ساختار درختی داده‌ها در فضای حافظه اصلی پهن می‌کنیم. معماری دام، نه به زبان برنامه‌نویسی خاصّی وابستگی دارد و نه به سکّوی برنامه‌نویسی ویژه‌ای، بلکه، به منظور اجراء و پیاده‌سازی آن باید از یک زبان برنامه‌نویسی سطح بالا همچون جاوا، سی‌شارپ، جاوا اسکریپت یا مشابه آن‌ها سود بجوییم. آنسوی رابط کاربر سند با مدلی شیءگرا نمایانده می‌شود.

۲-۲-۵: شئ XMLHttpRequest

برای هرگونه ارتباط با سرور در جاوا اسکریپت، بایستی از شیء XMLHttpRequest استفاده نمود. از آن جایی که این شیء جزء استانداردهای وب نیست می توانید، از جاوا اسکریپت به دو طریق برای ساخت آن استفاده کنید:

ساخت شیء XMLHttpRequest در مرورگر IE:

ساخت این شیء در نسخه های مختلف مرورگر IE نیز به دو صورت ساخته می‌شود:

```
var xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
```

یا

```
var xmlHttp = new ActiveXObject("Msxml2.XMLHTTP ");
```

ساخت شیء XMLHttpRequest در سایر مرورگرها

```
var xmlHttp = new XmlHttpRequest;
```

در حالت کلی و برای تمامی مرورگر ها می‌توان از قطعه کد زیر استفاده کرد:

```
var xmlHttp;
// Browser detection:
try { xmlHttp=new XMLHttpRequest(); }
catch (e)
{
   try { xmlHttp=new ActiveXObject("Msxml2.XMLHTTP"); }
   catch (e)
   {
      try {xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");}
      catch (e)
      {
        alert("Your browser does not support AJAX!");
        return false;
      }
   }
}
```

۲-۲-۵-۱: خصوصیات و توابع شیء XMLHttpRequest

حال که با ساختن این شیء XMLHttpRequest آشنا شدیم، به معرفی توابع و خصوصیات آن می‌پردازیم:

جدول ۲-۱: شرح خصوصیات شیء XMLHttpRequest

خصوصیت	شرح
onreadystatechange	اشاره گر به تابعی که باید هنگام تغییر حالت خصوصیت readystate اجرا شود
readystate	عددی که نشانگر وضعیت تقاضای ما به سرور است: عدد ۰ نمایانگر عدم آغاز ارتباط (unitialized) عدد ۱ نمایانگر حالت باز(open) عدد ۲ نمایانگر اتمام ارسال(sent) عدد ۳ نمایانگر حالت دریافت (receiving) عدد ٤ نمایانگر اتمام بارگذاری (loaded)
responseBody (IE7 Only)	پاسخ به صورت آرایه‌ای از بایت
responseText	پاسخ به صورت رشته متنی
responseXML	پاسخ به صورت شیء Dom Xml
status	کد وضعیت Html
statusText	متنی که status را شرح می‌دهد.

جدول ۲-۲: شرح توابع شیء XMLHttpRequest

تابع	شرح
abort	قطع تقاضای در حال پیشروی
getAllResponseHeaders	دریافت لیست کاملی از سروندهای Http تقاضا
getRespnoseHeader	دریافت سروند یک تقاضای Http خاص
open	دارای چند آرگومان است که اولی نوع تقاضا را مشخص می‌کند، بعدی آدرس URL مقصد را مشخص می‌کند و آرگومان آخر همزمان بودن(true) یا غیر همزمان بودن تقاضا(false) را مشخص می‌کند.
send	ارسال تقاضا به سرور
setRequestHeader	اضافه کردن یک سروند Http سفارشی به تقاضا

می‌توان دید که onreadystatechange نمایانگر کدی است که باید اجرا شود، تابع open برای ایجاد یک درخواست است و تابع send برای ارسال تقاضا می‌باشد. خصوصیت readystate به ما کمک می‌کند تا بدانیم کُد دقیقاً چه زمانی نسبت به پاسخ عکس العمل نشان دهد و responseText و responseXML برای بدست آوردن پاسخ به کار می‌رود. در پایان تابع abort برای خاتمه دادن به یک تقاضای در حال پیشروی است. اگر چه خصوصیات و توابع دیگری در دو جدول فوق وجود دارد، ولی ۷ خصوصیت و تابع ذکر شده اساس کاربردهای ما را تشکیل می‌دهد.

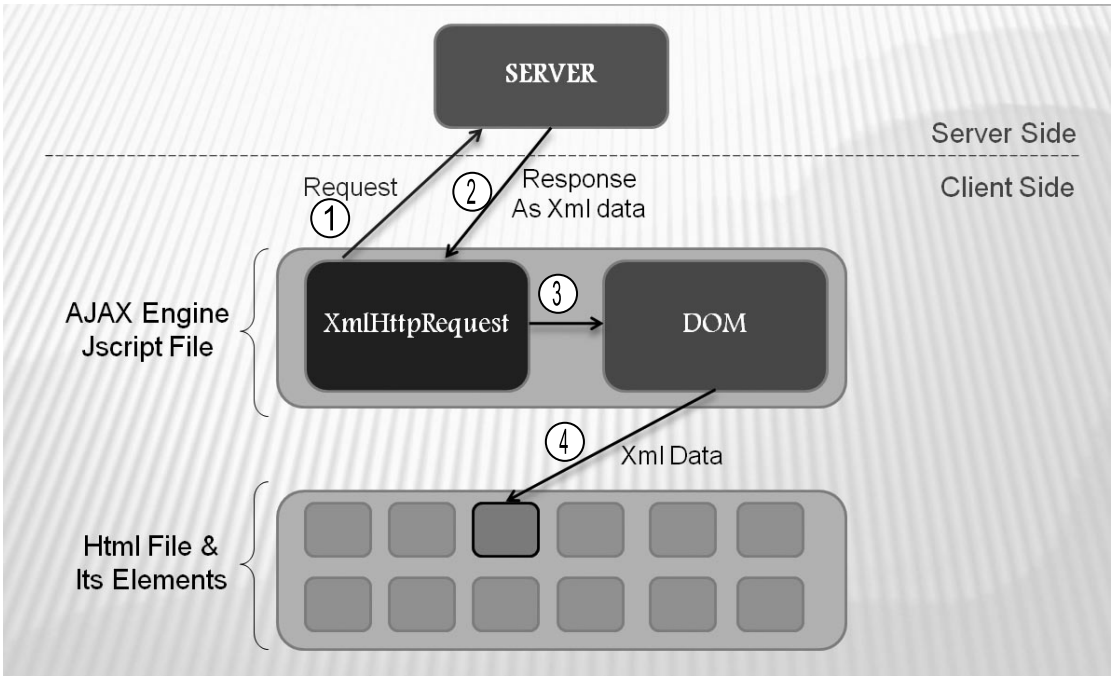
مثال

در قطعه کد زیر می‌بینید که در خط ۱ خاصیت onreadystatechange را به تابع getData نسبت می‌دهیم. سپس از تابع open استفاده می‌کنیم تا متن داخل inText را به آدرس مشخص شده بفرستیم. سپس با استفاده از تابع send تقاضای خود را ارسال می‌کنیم.

```
xmlHttp.onreadystatechange = getData;
xmlHttp.open("GET","Translate.aspx?inWord=star").value ,true);
xmlHttp.send(null);
```

۲-۳: چگونگی به کارگیری فناوری‌ها در AJAX

مراحل کلی پاسخ به تقاضاهای کاربر در AJAX همانند شکل (۲-۱) می‌باشد.
ابتدا کاربر درخواستِ خود را از طریقِ یک رویداد مانند کلیک بر روی یک دکمه اعلام می‌کند، سپس درخواستِ کاربر از طریق شیء XmlHttpRequest به سرور فرستاده می‌شود. سپس سرور پاسخ را به صورت داده Xml ارسال می‌کند. پاسخ دریافت شده توسط موتور AJAX (کدهای جاوا اسکریپت) پردازش شده و با استفاده از فناوری DOM قسمتی از صفحه Html که مورد نیاز است به روز می‌شود. بدین ترتیب ارتباط کاربر با نمای سایت همواره باقی خواهد ماند و از refresh های پی‌درپی جلوگیری می‌شود.



شکل ۲-۱: چگونگی به کارگیری فناوری‌های مختلف در AJAX

۲-۴: تفاوت مدل AJAX با مدل کلاسیک برنامه های تحت وب

هدف اصلی از ارائه مدل AJAX، نزدیک کردن برنامه‌های تحت وب به برنامه‌های رومیزی است. و در حالت کلی می‌توان گفت که برنامه‌های تحت وب در مدل AJAX نسبت به مدل کلاسیک به نرم افزارهای رومیزی نزدیک‌تر هستند. اما تفاوت‌های این دو مدل را می‌توان به صورت زیر بیان کرد.

۱. در مدل AJAX به جای بارگذاری صفحات ابتدا موتور AJAX (کدهای جاوااسکرپیت) بارگذاری شده و سپس نمای سایت توسط این موتور ساخته و بارگذاری می‌شود.

۲. در مدل AJAX رویدادها و تقاضاهای کاربر توسط موتور AJAX و شیء XmlHttpRequest به سرور ارسال می‌گردد. در نتیجه ارتباط کاربر با نمای سایت قطع نمی‌شود و refresh های متوالی وجود ندارد.
۳. در مدل AJAX در هر درخواست کاربر تنها قسمت‌های مورد نیاز از صفحه جاری به‌روز می‌شود و نیازی به بارگذاری مجدد قسمت‌های ثابت سایت وجود ندارد.
٤. در مدل AJAX می‌توان از امکانات گرافیکی بیشتر از جمله drag & drop و ... برای انجام امور استفاده نمود.

۲-۵: معایب استفاده از AJAX

۲-۵-۱: عدم کارآیی مناسب دکمه های Back و Forward

یکی از مسائلی که به عنوان ایراد مطرح شده، دشواری تغییر عادت کاربران در استفاده از کلیدهای Back و Forward و Refresh در مرورگرهای وب است. یکی از مشکلات برنامه نویسان وب همواره این است که یا باید کاربر را عادت دهند که هرچه کمتر از این دکمه‌ها استفاده کند و یا نرم‌افزار خود را طوری بنویسند که اگر کاربر سهوا یا عمدا از این دکمه‌ها استفاده کرد، نرم افزار دچار اشتباه و خطا در تفسیر عمل کاربر نشود.

به عنوان مثال هنوز بسیاری از سایت های تجارت الکترونیکی که به کار فروش محصولات مشغولند، هنگام طی شدن مراحل نهایی خرید آنلاین به کاربر هشدار می دهند که حین پردازش یک سفارش (یعنی درست در لحظه‌ای که فرمان نهایی از سوی کاربر ارسال شده است و هنوز صفحه نمایش پیغام ثبت موفقیت آمیز سفارش یا عدم ثبت آن برای وی نمایش داده نشده) از فشردن کلید Refresh جدًا پرهیز کنند وگرنه ممکن است از کارت اعتباری آن‌ها دوبار پول کسر شود. همچنین استفاده از دکمه های Back و Forward در عملیاتی که به آسانی برگشت پذیر نیستند، ممکن است باعث گیج شدن کاربر شوند.

مثلا اگر نامه های داخل صندوق پستی خود را پاک کنید، استفاده از دکمه Back هرگز این عمل را Undo نمی کند. همچنین اگر یک قلم کالا به سبد خرید آنلاین خود اضافه کنید، فشردن دکمه Back ممکن است در ظاهر چنین نشان دهد که آن قلم کالا مجددا از سبد برداشته شده، اما در سمت سرور همچنان در سبد خرید کاربر باشد. از آنجا که فناوری

ای‌جکس عمل Navigation یا راهبری در یک سایت را به روندی غیر خطی تبدیل می-کند، تمام این مشکلات به شکل حادتری ممکن است بروز کند.
در واقع با حضور Ajax، کارکرد سیستم History مرورگر به مساله ای بغرنج تبدیل می شود. زیرا برنامه نویس یا باید با گنجاندن دکمه ها و فرامین اضافی، مکانیزم Undo را بازسازی کند و یا موتور ای‌جکس را طوری بنویسد که فشردن دکمه Back خود به خود موجب احضار فرمان Undo شود. در هر دو صورت کار برنامه نویس آسان نخواهد بود. البته در این زمینه ترفندها و تکنیک‌هایی هم ابداع شده است. از جمله، استفاده از تگ IFRAME مخفی در اینترفیس صفحه که موتور ای‌جکس بتواند در صورت فشرده شدن دکمه Back از سوی کاربر، نسخه های پیشین نمایش داده شده از اینترفیس را از انباره History مرورگر بیرون بکشد و دوباره در چرخه عملیات موتور ای‌جکس وارد کند. این ترفند هم اکنون در سرویس Google Maps استفاده می‌شود.

۲-۵-۲: وابستگی به XmlHttpRequest

یک مشکل عمده دیگر نیز در ارتباط با ای‌جکس وجود دارد. این فناوری به شدت متکی به XMLHttpRequest است و این شیء به دلایل امنیتی طی ماه های اخیر هرچه بیشتر و بیشتر در نسخه‌های جدید مرورگرها محدود شده است. زیرا اگر هر کلاینتی بتواند از هر نقطه‌ای به هر سروری این فرمان را بفرستد، آنگاه تهدیدهای امنیتی علیه سایت‌ها افزایش می‌یابد. محدودیت‌های جدید اعمال شده در نسخه های اخیر مرورگرها موجب شده که فرمان XMLHttpRequest به غیر از سایتی که صفحه وب از آنجا آمده است نتواند با سایت دیگری ارتباط داشته باشد و این مسئله در تناقض با کاربرد ای‌جکس در زمینه وب‌سرویس است. البته برای غلبه بر این مشکل راه حل‌هایی هم پیشنهاد شده است، از جمله این که شیء XMLHttpRequest می‌تواند تقاضای ارتباط با سایت های دیگر را به یک وب‌سرویس روی سایتی که صفحه وب از آنجا آمده است، بفرستد و این وب-سرویس (که روی میزبان سایت قرار دارد و با تمام اینترنت در ارتباط است) به صورت یک واسطه عمل کند و تقاضاهای مورد نظر را برای سایت مقصد ارسال کند. این وب-سرویس های واسطه اصطلاحا Application Proxy نامیده می‌شوند.

البته ای‌جکس مشکلات کوچک و جنبی دیگری هم دارد که چندان مایه نگرانی نیست ولی به هرحال قابل لمس هستند. به عنوان نمونه، نرم افزارهای مبتنی بر ای‌جکس از حجم زیادی جاوا اسکریپت استفاده می کنند، که همه این‌ها در هر بازدید دست کم یک بار باید

روی مرورگر بارگذاری شوند. بنابراین اولین باری که چنین اینترفیسی بارگذاری می-شود، صفحه وب آن قدر سنگین می‌شود که حتی با ارتباط پرسرعت هم چند لحظه طول می کشد که صفحه بارگذاری شود.

در چنین شرایطی نوشتن یک موتور ای‌جکس هوشمند که با میزان کد کمتر بتواند بهترین کارکرد را داشته باشد، خود به یک چالش برنامه نویسی تبدیل می‌شود؛ ضمن اینکه کاربران وب در کشورهایی که سرعت دسترسی به اینترنت در آنها به طور معمول زیاد نیست، باید هنگام بارگذاری اینگونه صفحات وب صبر پیشه کنند و برنامه نویسان نیز مراقب باشند تا در صورتی که به دلیل کندی خط یا قطع شدن‌های لحظه‌ای آن، کدهای جاوا اسکریپت به طور کامل روی مرورگر بارگذاری نشد، نرم افزار دچار خطا و اشتباه نشود و بتواند این مشکلات را از طریق بارگذاری مجدد و هوشمندانه کد جاوا اسکریپت روی کلاینت، مدیریت کند.

فصل ۳

پیاده سازی AJAX

در این بخش یک نمونه از پیاده سازی AJAX را بررسی می‌کنیم. برای سادگی کار از تولید نمای سایت توسط موتور AJAX خودداری می‌شود. و فقط ارتباط با سرور از طریق شیء XmlHttpRequest و استفاده از پاسخ آن را بررسی خواهیم کرد. از صفحات ASPX و زبان C# (با استفاده از VisualStudio2005) هم به عنوان فناوری سمت سرور استفاده خواهیم کرد.

قصد ما ساخت یک دیکشنری تحت وب است.(همانند شکل ٤-١) در این دیکشنری لحظاتی پس از تایپ کلمه توسط کاربر(بدون نیاز به فشردن دکمه یا کلید Enter) معنای آن لغت نمایش داده می‌شود، بدون این که صفحه refresh شود.

فایل‌های مورد نیاز به شرح زیر هستند.

AjaxDictionary.htm و Style.css: فایل HTML (نمای سایت) و فایل CSS،

ajax.js: کدهای جاوا اسکریپت(سمت کلاینت)،

Translate.aspx و Translate.aspx.cs: فایل‌های سمت سرور،

و **فایل Dictionary.mdb در پوشه App_Data**: پایگاه داده شامل لغات و معانی.

پس از ساختن این فایل‌ها بایستی در VisualStudio 2005 یک وب‌سایت جدید ساخته و این فایل‌ها را به پروژه خود اضافه نمایید.

شکل ۳-۱: دیکشنری ساخته شده با تکنیک AJAX

۳-۱: فایل AjaxDictionary.htm:

```
<html>
<head>
    <title>Ajax Dictionary </title>
    <link rel="stylesheet" type="text/css" href="Style.css" />
</head>
<body onload="">

    <script type="text/javascript" src="ajax.js"></script>

    <form id="Form1" class="form" >
        <img id="ban" src="ban.jpg">
        <hr></hr>
        Word:
        <input id="inText" name="inText" type="text"
onkeyup="setTimeout('ajaxFunction()',500);" /><br />
        <img id="loading" src="loading.gif" style="visibility:
hidden" /><br />
        Translation:<br />
        <textarea id="transTxt" class="trans" dir="rtl"></textarea>
        <hr></hr>
    </form>
    Designed by Sajjad Rezaee - sajjad.rezaee@yahoo.com<br />
    <span style="font-size: 8pt">Database from Babylon </span>
</html>
```

قسمت‌های مهم این فایل عبارتند از:

خطِ

```
<link rel="stylesheet" type="text/css" href="Style.css" />
```

که فایل Style.css را به فایل Html پیوند می‌زند.

خطِ

```
<script type="text/javascript" src="ajax.js"></script>
```

که فایل ajax.js را به فایل Html پیوند می‌زند.

قسمتِ

```
<input id="inText" name="inText" type="text"
 onkeyup = "setTimeout ('ajaxFunction()',500);" />
```

که محل وارد کردن کلمه ورودی است. قسمت id برای دسترسی به ورودی با استفاده از DOM است. قسمت onkeyup وقتی که کاربر کلیدی را می‌فشارد با تاخیر ۵۰۰ میلی ثانیه تابع ajaxFunction را از فایل جاوا اسکریپت اجرا می‌کند.

قسمتِ

```
<img id="loading" src="loading.gif" style="visibility: hidden" />
```

تصویری را برای زمانی که سرور در حال پردازش درخواست ماست، نمایش می‌دهد.

و قسمتِ

```
<textarea id="transTxt" class="trans" dir="rtl"></textarea>
```

محل نمایش خروجی است که با شناسه transTxt مشخص می‌گردد.

۳-۲: فایل Style.css:

```
.form
{
	background-color:lavender;
	width: 490px;
	height: 368px;
}
.trans
{
	width: 100%;
	height: 120px;
	font-family :Tahoma;
}
```

دو کلاس form و trans در این فایل با خصوصیات جداگانه‌ای تعریف شده‌اند. در فایل Html هر قسمتی که خصوصیت class در آن‌ها تعریف شده باشد، دارای خصوصیات آن کلاس در فایل CSS خواهند بود. برای مثال در فایل Html ما قسمت:

```
<form id="Form1" class="form" >
```

دارای خصوصیات کلاس form از فایل CSS خواهد بود.

۳-۳: فایل ajax.js:

```
function ajaxFunction()
{
  var xmlHttp;
  // Browser detection:
  try { xmlHttp=new XMLHttpRequest(); }
  catch (e)
  {
     try { xmlHttp=new ActiveXObject("Msxml2.XMLHTTP"); }
     catch (e)
     {
        try {xmlHttp=new ActiveXObject("Microsoft.XMLHTTP");}
        catch (e)
        {
          alert("Your browser does not support AJAX!");
          return false;
        }
     }
  }
  // End of browser detection

  xmlHttp.onreadystatechange = getData;
  var inText = document.getElementById("inText").value;
  xmlHttp.open("GET","Translate.aspx?inWord="+inText,true);
  xmlHttp.send(null);


  function getData()
    {
      if((xmlHttp.readyState > 0)&&(xmlHttp.readyState < 4))
      {
      document.getElementById("loading").style.visibility =
'visible';
      }
      if(xmlHttp.readyState==4)
      {
      document.getElementById("transTxt").value=xmlHttp.responseText;
      document.getElementById("loading").style.visibility = 'hidden';
      }
    }
}
```

این فایل که یکی از قسمت‌های اصلی مثال ما به حساب می‌آید، کار ارتباط پشت صحنه با سرور و دریافت معنای لغات و درج آن در محل خروجی را برعهده دارد.

در ابتدای فایل کدهای تشخیص مرورگر را می‌بینیم که قبلاً راجع به آن صحبت کرده‌ایم.
پس از اینکه شیء xmlHttp در این قسمت ساخته شد، می‌توانیم از آن استفاده کنیم.
در قطعه کدِ

```
xmlHttp.onreadystatechange = getData;
var inText = document.getElementById("inText").value;
xmlHttp.open("GET","Translate.aspx?inWord="+inText,true);
xmlHttp.send(null);
```

ابتدا خاصیت onreadystatechange شیء xmlHttp را به تابع getData که راجع به آن صحبت خواهد شد، مقداردهی می‌کنیم. در خط دوم سپس متغیر inText را با استفاده از فناوری DOM با لغت وارد شده مقداردهی می‌نماییم. در دوخط بعد تقاضای خود را از طریق توابع open و send به سرور ارسال می‌کنیم و منتظر دریافت پاسخ می‌شویم.
در تابع getData هر گاه که خاصیت readyState با مقدار ٤ (اتمام بارگذاری پاسخ) مساوی شد، پاسخ را در خروجی که با شناسه transTxt مشخص می‌شود، نمایش می‌دهیم.

۳-۴: فایل Translate.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="~/Translate.aspx.cs"  Inherits="Translate" %>

<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
    <title>Dictionary Sample</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        </div>
    </form>
</body>
</html>
```

این فایل یک فایل ساده Aspx است که برای دریافت معنی لغت از آن استفاده می‌کنیم.
قسمت مهم آن خط اول آن است.

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeFile="~/Translate.aspx.cs"  Inherits="Translate" %>
```

خصوصیت CodeFile در این قسمت ارتباط با فایل Translate.aspx.cs را که کدهای سمت سرور ما در آن قرار دارد، برقرار می‌کند.

۳-۵: فایل Translate.aspx.cs:

```
using System;
using System.Data;
using System.Configuration;
using System.Collections;
using System.Web;
using System.Web.Security;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.WebControls.WebParts;
using System.Web.UI.HtmlControls;
using System.Data.OleDb;

public partial class Translate : System.Web.UI.Page
{
  protected void Page_Load(object sender, EventArgs e)
  {
    Response.Clear();
    string inWord = Request.QueryString["inWord"];//get input word
    //from address
    string outWord = GetTrans(inWord);//call GetTrans() & get
    //translation of input word
    Response.Write(outWord);// send translation to client
    Response.End();
  }
  private string GetTrans(string inWord)
  {
    try
    {
      //get translation of word from Dictinary Database:
      string cs = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=|DataDirectory|\\Dictionary.mdb";
      OleDbConnection oc = new OleDbConnection(cs);
      oc.Open();
      OleDbDataAdapter od = new OleDbDataAdapter("Select * From Word
where [in] Like '" + inWord + "'", oc);
      DataSet ds = new DataSet();
      od.Fill(ds);
      return ds.Tables[0].Rows[0][1].ToString();
      oc.Close();
    }
    catch { return "هیچ موردی پیدا نشد"; } //word dosen't found
  }
}
```

تابع Page_Load قبل از بارگذاری صفحه اجرا می‌شود.

```
protected void Page_Load(object sender, EventArgs e)
{
  Response.Clear();
  string inWord = Request.QueryString["inWord"];//get input word
  //from address
  string outWord = GetTrans(inWord);//call GetTrans() & get
  //translation of input word
  Response.Write(outWord);// send translation to client
  Response.End();
}
```

در خط اول، تابعِ Clear از شیء Response فراخوانی شده است. شیء Response این امکان را به ما می‌دهد تا داده‌ها را از سرور به کلاینت ارسال کنیم. تابع Clear بافرها را خالی و آماده برای ارسال داده‌های جدید می‌کند.
در خط دوم متغیر inWord را با مقداری که در آدرس URL و قسمت inWord قرار دارد، مقداردهی اولیه می‌کنیم. به عنوان مثال اگر صفحه با آدرسِ
http://www.sample.com/translate.aspx?inWord=**star**
فراخوانی شود، متغیر inWord دارای مقدار "star" خواهد شد.
در خط بعد متغیر outWord را با مقدار بازگشتی از تابع GetTrans مقداردهی می‌کنیم. تابع GetTrans که بعد از تابع PageLoad تعریف شده است، یک کلمه را به عنوان آرگومان گرفته و پس از اتصال به پایگاه داده لغات معنی آن را برمی‌گرداند. یعنی پس از اجرای این خط متغیر outWord دارای معنای کلمه ورودی می‌باشد.
در خط بعد outWord را با استفاده از شیء Response برای کلاینت ارسال می‌کنیم.
در خط آخر تابع Page_Load هم از تابع End برای اتمام ارسال استفاده می‌کنیم.

۳-۶: فایل Dictionary.mdb:

این فایل پایگاه داده Access است که دارای یک جدول به نام Word است. جدول Word دارای دو ستون in و out است که به ترتیب کلمات و معانی در این دو ستون قرار خواهند گرفت.

پس از ساختن این فایل‌ها و اضافه کردن آن‌ها به پروژه وب‌سایت خود در VisualStudio 2005 می‌توانید فایل Dictionary.htm را در محیط نرم افزار VisualStudio و یا با استفاده از نرم افزار IIS اجرا نمایید و نتیجه را مشاهده نمایید.

فصل ۴

کاربردهای AJAX

از زمان ظهور پدیده AJAX در دنیای وب، سایت‌ها و شرکت‌های زیادی اقدام به استفاده از این نوع معماری در صفحات وب خود نموده‌اند، که شرکت Google در این امر پیشتاز بوده است و قبل از نامگذاری AJAX از این تکنیک استفاده می‌کرده است.
از جمله کاربردها و سایت‌های استفاده کننده از AJAX می‌توان موارد زیر را نام برد:

- **چت**
  - سرویس چت گوگل
- **تجارت الکترونیکی**
  - سایت closo.com
  - سایت جستجوی خرید آمازون (A9.com)
- **بازی**
- **پیام رسانی اینترنتی**
  - سایت membo.com
- **پست الکترونیکی**
  - جی میل(Gmail.com)
  - یاهو میل (Yahoo! Mail)
- **نقشه های آنلاین**
  - Google Maps

- **سیستم عامل های تحت وب**
  - سیستم عامل تحت وب AjaxWindows
  - سیستم عامل تحت وب eyeOS
- **مدیریت اطلاعات شخصی**
  - سایت مدیریت پروژه BaseCamp
- **جستجو گرها**
  - Google Suggest
  - Yahoo! Instant Search
- **پورتال**
  - سرویس پورتال گوگل(Google Personal Homepage)
- **نرم افزار های اداری تحت وب**
  - نرم افزار تحت وب پردازش متن Writely
  - نزم افزار صفحه گسترده تحت وب Google Spread Sheets

فهرست منابع

منابع فارسی:

۱. صادقی جابر، اصول تکنیک آژاکس، نسخه اینترنتی، ۱۳۸٦.

۲. نوعی پور بهروز، "همه چیز درباره Ajax"، ماهنامهٔ شبکه، شماره ٦۲، بهمن ۸٤

منابع خارجی

1. Chris Ullman, Lucinda Dykes, Beginning Ajax, Wiley Publishing Inc., Indianapolis, 2007.
2. Jesse James Garrett, "Ajax: A New Approach to Web Applications", Adaptive Path, [online], Available: http://www.adaptivepath.com /publications/essays/archives/000385.php, 2009.
3. Saad Hamid, "Web 1.0 vs Web 2.0, the Difference", Sizlopedia, [online], Available: http://www.sizlopedia.com/2007/08/18/web-10-vs-web-20-the-visual-difference/, 2009.