دلایل ظهور سیستمهای

Application Server

دلایل ظهور سیستم های Application server :

دلیل توسعه و بسط این سیستم ها در پاسخگویی گروههای عظیمی از کاربران نهفته است. توسعه دهندگان احتیاج به ابزاری داشتند که بتواند کلیه منابع سطح پایین مورد استفاده برنامه های کاربردی اینترنت را در قالبی منسجم و قابل استفاده کپسوله کند. به گونه ایی که ایجاد برنامه های کاربردی محاوره ایی و کار آمد، برای اینترنت به راحتی ممکن شود.

از این دیدگاه سیستم های Application server شباهت بسیاری به سیستم عامل ها دارند چرا که سیستم عامل های امروزی با ارائه ابزارهای کار آمد خود امکانات مورد نیاز جهت انجام کارهای مختلف را در اختیار کاربران خود قرار می دهند.همان گونه که سیستم عاملها، مجموعه ایی از اسباب های سخت افزاری و سرویس های مورد استفاده جهت نمایش پنجره ها را کپسوله می کنند، سیستم های Application server نیز انواع پروتکلها و داده ها و همچنین مجموعه های مختلفی از سرویسهای مورد استفاده جهت برقراری ارتباط با کاربران را فراهم می کند. سیستم های Application server همانند برنامه های دیگری که زیر بنای آنها دائما" از سوی توسعه دهندگان در حال تغییر و تحول است، ناچار به دگرگونی در مقابل نیازهای آنان هستند.

همانند تغییراتی که در دهه گذشته به دلیل ظهور نیازهای جدید، سیستم عاملها پـذیرای آنها بودند، محیطهای تجارت الکترونیکی و اینترنتی نسل آینده، موجب تحولی عظـیم در سیستم های Application server خواهد شد. برای درک بهتر اهمیت این سیستم ها بحث خود را بررسی تاریخچه این ابزارها بکار آمد پی می گیریم.

تاریخچه سیستم های Application server :

توسعه نرم افزاری بر روی اینترنت از قدمت چندانی برخوردار نیست، با این وجود طـی همین مدت کوتاه به دلایل رویدادهای مهمی که در زمینه اینترنت رخ داده است، توسـعه نرم افزاری به توسعه دهندگان سایت کرده است.

هدف توسعه دهندگان وب از تبدیل صفحات ایستایHTML به صفحات دینامیک، بـه توسعه برنامه های کاربردی کارآمد بر روی وب تکامل یافته است. به واسطه این تکامـل بـه تـدریج چـار چـوب هـای سیسـتمی کـه امـروزه بـه سیسـتم هـای application serverمعروف شده،تعریف و به مرور زمان تکمیل شد. اما ابتدا به تکنولوژی وابزارهای مختلفی که موجب مطرح شدن این گونه سیستم ها شدند نگاهی کوتاه بیندازیم.

CgIاولین تکنولوژی بود که امکان ایجاد صـفحات html پویـا را در اختیـار طراحـان قرارداد. متاسفانه بهره گیری از این تکنولـوژی کارزیـادی را از جانـب توسـعه دهنـدگان طلب میکـرد، چـرا کـه ایشـان عـلاوه بـر کـد نویسـی قـوانین کـار در قالـب صـفحات HTMLمجبور بودند تا به منظور انجام کارهای ساده مانند ارتباط با یک بانک اطلاعاتی

وقالب بندی نتایج به صورت HTMLو حفظ اطلاعات مابین جلسات (نوبت های مختلف برقراری ارتباط با سرور)راه حل منحصر به فردی را پیاده سازی کنند.

از آنجایی که بیشتر کاربران با مسائل مشترکی دست به گریبان بودند، توسعه دهندگان به ایجاد کتابخانه هایی متشکل از کدهای قابل استفاده مجدد اقدام کردند که بدین ترتیب بسیاری از مسائل حل می شد. با وجود این با پیچیده ترشدن برنامه ها،سرهم بندی این قطعات (کدهای قابل استفاده مجدد)در قالب یک برنامه کاربردی کامل و قابل اعتماد به تدریج مشکل ومشکلتر می شد.در آن مقطع برخی از دست اندر کاران امرکامپیوتر متوجه این نکته مهم شدند و بدین ترتیب سیستم های Application Serverپا به عرصه ظهورگذاشتند. این پیشگامان درابتدا اقدام به تدوین مجموعه ایی از منابع مفیدوقابل استفاده مجدد کردندند و آنها را در اختیار توسعه دهندگان پیش از این از آنها استفاده می کردند- بسیار مفید بودند. در ابتدای امر بیشتر این فرآوردها همانند ابزاری سودمند به توسعه دهندگان وب فروخته شد.همانند وضعیتی که پیش از این در مورد ابزارهای مورد استفاده جهت توسعه برنامه های کاربردی وجود داشت، در اینجا منابع مورد نیازحین اجرای برنامه از طریق یک زبان Scriptو یا به واسطه مجموعه ای از امکانات کپسوله شده (درقالب کلاس) فراهم می شد. نخستین پیشگامان در این عرصه از تکنولوژی، شرکت Allairبا نرم افزارColdfusion،شرکت Net Scapeبا نرم افزار Live Wireبودند.

در این هنگام تب وب در میان کاربران عادی و تجاری همه گیرشده و به واسطه برنامه های کاربردی وب ، مجال بسیار مناسبی برای بهره گیری ازاین ابزارها فراهم شد. در نتیجه شمار بیشتری ازتوسعه دهندگان ، اقدام به توسعه برنامه هایی کردندکه قابلیتهای جامع تری را نسبت به آنچه که پیشتر تولید می کردند(صفحات پویای وب)در اختیار می گذاشتند. برای پاسخ به این نیازها تولید کنندگان نرم افزار شروع به گسترش محوده فعالیت های خود کردند.

آنها برای گسترش زمینه کاری و افزایش کارایی نرم افزارهایشان از تکنولوژی TP monitorsنیز پیشیبانی کردند. برای تسهیل در استفاده از نرم افزارهای تولید شده به همراه سیستم های موجود ،ابزارهایی را جهت تطبیق با سیستم های E- mailو کاربا ORBها (جهت اتصال با برنامه های کاربردی موجود) توسعه دادند. این گونه برنامه ها به واسطه دارا بودن ابزارهایی جهت اعتبار سنجی و رمز گذاری از امنیت بالایی برخوردار بودند. به موازات توسعه این قبیل برنامه های کاربردی ، اهمیت فوق العاده آنها در تجارت الکترونیکی بیش از بیش برای دست اندر کاران کامپیوتر مشهود می شد.

همزمان با ظهور این گونه برنامه های کارآمد ، شرکتهای فعال در این زمینه ، به امر بهبود ابزارها و زبان برنامه نویسی که دسترسی به منابع مذکور را ممکن می ساخت، سرعت بخشیدند. با این هدف که کار توسعه برنامه ها پربارتر شود. نتیجه این فعالیت رشد فزاینده زبانهایScriptبود.

با وجود اینکه هنوز اصطلاح سیستم های Application Serverوارد واژگان اینترنتی ما نشده بود هر یک از این سیستم های مسیر تکامل خود را به طور جداگانه طی می کردند ، تقریبا" در اواسط سال ۱۹۹۹بیشتر آنها برسر ساختار یکسان توافق کرده بودند. این ساختار شامل مجموعه ایی از سرویسهای حین اجرا و همچنین سرویس های مورد نیاز جهت اتصال به سایر سیستم ها و منابع (بانکهای اطلاعاتی ، فایل سرورها، خدمات پست الکترونیکی) و سایر برنامه های کاربردی می شد

برای دسترسی به این منابع، توسعه دهندگان یک مدل دو لایه ایی در اختیار داشتند. لایه اول در این مدل که Application Layerنام داشت ، عهده دار نمایش اطلاعات به کاربران بود و دیگری با نام Business Logic Layer، فرایند دسترسی به داده ها و پردازش آنها را انجام می داد با همه گیر شدن تدریجی این مدل ، اصطلاح Application Serverنیز برسرزبانها افتاد.

ضرورت ایجاد استانداردها :

طی دوسال گذشته در اثر حرکت شرکتهای توسعه دهنده سیستم های Application Server به سمت استاندارد کردن ساختار آنها و نیز تدارک رابطهایی که از آن طریق ، سایر توسعه دهندگان بتوانند به منابع موجود بر روی سرورها، دسترسی پیدا کنند، این بخش از صنعت نرم افزار دستخوش تغییرات بسیاری شد .

در ابتدا این استانداردها به صورت مجزا مطرح شدند. برای نمونه ابزار Java Server Pagesبه عنوان دو ابزار بسیار کار آمد جهت استفاده از جاوا در لایه اول از مدل دو لایه

ایی ذکر شده در قسمت قبل مورد توجه بسیاری از توسعه دهندگان قرار گرفتند. از طرف مایکروسافت با ابزار Aspو شرکت All air با محصول cfmlابزارهای دیگری را جهت استفاده در این لایه معرفی کردند.

به طور مشابه دو تکنولوژی Ejb از شرکت Sun MicroSystem و com /com + از شرکت مایکرو سافت به عنوان استانداردهایی به توسعه لایه دوم مطرح شدند.

در گام بعدی SUNو سایراعضای جامه جاوا سعی کردند تا این استانداردهای مجزا را در قالبی همگن و مجموعه ی کاملی از API ها که هم اکنون آنرا با نام java 2EE میشناسیم ، گردآوری کنند.

تا زمانی که حرکت به سمت استاندارد کردن مدلهای برنامه نویسی ادامه دارد، این وضعیت وفعالیت توسعه دهندگان دستخوش تغییرات شگرفی خواهد شد. مشابه حرکتی که در مورد استاندارد کردن WIN32 Apiصورت گرفت و موجب ظهور برنامه های کاربردی کارآمد تری شد، در اینجا نیز وجود مجموعه همگنی از ابزارهای برنامه نویسی این امکان را در برابر تولید کنندگان نرم افزار را مجبور خواهندکرد، هزینه حمل برنامه های کاربردی از یک سرور دیگر را کاهش دهند.

با این وجود قابل حمل نبودن برنامه های کاربردی ، مزیت بزرگی برای توسعه دهندگان نرم افزارها قرار می دهد تا بتوانند بسته های نرم افزاری قابل حملی را بر روی سیستم هایApplication Serverمختلف ایجاد کنند. چنانچه ایده قابل حمل بودن برنامه ها

برروی این سیستم های متفاوت کاملا تحقق نیابد، تولید کنندگان نرم افزار مجبور خواهند شد ، هزینه حمل برنامه های کاربردی ازیک سرور دیگر را کاهش دهند.

با این وجود قابل حمل نبودن برنامه های کاربردی ، مزیت بزرگی برای توسعه دهندگان محسوب می شود . چرا که توسعه برنامه تحت سرورها و انواع سیستم های عامل را تجربه می کنند.

متاسفانه استاندارد کردن نیز معایبی دارد. برای نمونه از آنجایی که تکنولوژی J2EEجزییات سیستم های عامل را پنهان نگاه می دارد. برنامه های کاربردی ساخته شده را با استفاده از این تکنولوژی اغلب از کلیه سرویسهای غنی ارائه شده توسط سیستم عامل های امروزی بهره کافی نمی گیرد.

علاوه براین تکنولوژی J2EE مدل پیچیده ایی برای توسعه دهندگان محسوب می شود. تسلط بر کلیه ابزارهای این تکنولوژی توسط تنها یک توسعه دهند اغلب مستلزم تلاش بسیاراست و این کار تنها از عهده توسعه دهندگان حرفه ایی بر می آمد. چنین می توان نتیجه گرفت که یکی از کارها ی مهمی که توسعه دهندگان سیستمهای Application Server از ابتدای ظهورشان با آن مواجه بودند و قابلیت به کارگیری آنها به همراه سایر تکنولوژی هایی است که ابزارها به یک توسعه دهنده جهت توسعه برنامه ها محسوب می شوند.

برای نمونه با متداول شدن روز افزون سیستم های قابل حمل (همانند تلفن همراه) و افزایش تصاعدی تعداد اتصا لات ، کاربران این ابزارها خواهان دسترسی به سرویسهایی

هستند که درحال حاضر از طریق وب قابل استفاده است. با این وجود،آنهایی که این گونه سیستم ها دارای رابطهای کاربردی متفاوتی بوده و پهنای باند آنها به اندازه ایی نیست که بتوان از آنها در یک مرورگر وب استفاده کرد ، نیاز به تکنولوژی های مختلف در لایه Web Application S erverمطرح می شود .

مزایای سیستم های Application Server

صفحات سرویس دهنده فعال (Active Server Pages) جدید ترین تکنولوژی در زمینه سرویس دهنده می با شند، که توسط شرکت ماکروسافت برای ایجاد صفحات HTMLمحاوره ایی و پویا در سایت وب جهانی و شبکه های داخلی طراحی شده است.

با استفاده از این تکنولوژی به راحتی می توانید با پایگاه داده هایی که در سرویس دهنده ها قرا ردارند ارتباط برقرار کنید. بنا به نیاز خود آنها را تغییر دهید: دادهای خود را با کاربران دیگر به اشتراک بگذارید.

همچنین می توانید خود را به سایتهای گوناگون معرفی کرده، علایق خود را به آنها بشناسید و اطلاعات مورد نیاز خود را از آنها در یافت نمایید. ASP ها با شماری از اجزاء اکتیوایکس روی سرویس دهنده استاندارد دسته بندی شده. این اجزاءبه شما اجازه کارهایی مثل تصمیم گیری در مورد قابلیتهای مرورگر های مختلف یا گنجاندن یک شماره در صفحه وب را می دهد.

شما نباید خود را فقط به اجزاء استاندارد اکتیوایکس محدود کنید، هر چند که این اجـزاء بسیار مفید می باشند. شما می توانید اجزاء الحاقی اکتیوایکس را برای خود ایجـاد کنیـد. این بدان معنا است که هیچ محدویتی در چگونگی توسعه ASP برای شما وجود ندارد.

با استفاده از اشیاء تعبیه شده قابل دسترس دریک ASPشما می توانیـد اسـکریپت هـای خود را بسیار قویتری کنید دربین چیزهای دیگری این اشیاء بـه شـما اجـازه دریافـت و ارسال اطلاعات به مرورگر یا از آن را میدهد. برای مثال با استفاده از شیءRequestمـی توانید اطلاعاتی را یک کاربر با فرم HTMLارسـال کـرده دریافـت و بـه آن اطلاعـات توسط یک اسکریپت پاسخ دهید.

همچنین با بکارگیری اسکریپتهای روی سریس دهنده یک شما می توانید صفحات وبی با اجزاءپویا ایجاد کنید. به عنوان یک نمونه بسیار ساده شـما مـی توانیـد صـفحه وب را ایجاد کنید که هریک پیغام جدید یا تاریخ آن روزرا نمایش دهد.

می توان بدون توزیع بانک اطلاعاتی در سطح شبکه از طریق این صفحات و بـا اسـتفاده از وب سرور با بانک اطلاعـاتی محلـی ارتبـاط برقـرار نمـوده، وتحـت کنتـرل ایسـتگاه سرویس دهند اطلاعات را در اختیار متقاضی قرار داد. برای این منظوربا استفاده از شـیء کوکی می توان متقاضی صـفحه ASPرا شناسـایی نمـود و براسـاس هویـت وی سـطح دسترسی به اطلاعات را مشخص نمود.

این عمل در بانکهای توزیعی به سادگی قابل اجرا نیست. نکته قابل توجه در اینجاست که بانک اطلاعاتی کاملا به صورت محلی نصب می شود اما در عمل همانند بانک توزیعی دسترسی می گردد. گلوگاه دسترسی فرم ASPمی باشد.

میتوان با استفاده از امکان هدایت تقاضای متقاضی های صفحاتASP ، تقاضای هر متقاضی را به ایستگاه مناسب انتقال داد. برای نمونه فرض کنید متقاضیان با یک سرویس دهنده وب به عنوان سرویس گیرنده در ارتباط می باشند. متقاضی خواهان مشاهده فیش حقوقی خویش است. فیشهای حقوقی در سرویس دهنده های متفاوت محاسبه میشوند. سرویس دهنده ها در سطح وب می بایست در یافت شود.

به عبارت دیگر فرم اولیه که در آن مشخصات متقاضی فیش حقوقی قرار داده می شود. در داخل این فرم اطلاعات متاقضی شامل نوع فیش حقوقی و شماره کارمندی وارد شده و برای سرویس دهنده ارسال میشود. اکنون سرویس دهنده بر اساس نوع تقاضای متقاضی یک دستگاه کاری مشخص نموده و تقاضا را که دریافت فرم دوم جهت مشاهده فیش حقوقی است را ارسال می نماید. برای این منظور از دستور العمل Redirect Response می توان استفاده نمود.

پس از بکارگیری اکتیوایکس شکل توانمندی از ASP ها می باشد. با ایجاد ASPهایی که میتوانند با یک پایگاه دادوستد کنند، شما قادرخواهید بودکه سایتهای وب بسیار پیشرفته ایی را ایجاد کنید.

توزیع بار کار سرویس دهنده:

از آنجائیکه تعداد متقاضیان سرویس دهنده های وب می تواند بیشمار و غیر قابل پیش بینی باشد لذا، در این بخش تکنیکی برای طرح وپیاده سازی برنامه ها برروی وب ارائه می شود.که تاحدی بارکار سرویس دهنده های وب را می توان با استفاده از این نکنیک کاهش داد.

می توان با استفاده از امکان هدایت تقاضای متقاضی های صفحات ASP، تقاضای هر متقاضی را به ایستگاه مناسب انتقال داد.

ایستگاه مناسب در واقع ایستگاه ریاست که از لحاظ با رکاری سربار کمتری دارد.در اینجا برای مثال صفحه اول از روی سرویس دهنده به متقاضی داده می شود در این صفحه متقاضی صفحه بعدی ASPرا انتخاب می کند. سرویس دهنده براساس وضعیت سرویس دهنده های دیگری ازلحاظ میزان با رکاری تصمیم می گیرد که صفحه دوم را از کدام سرویس دهنده وب برای متقاضی باید ارسال نمود.

این سرویس دهنده های وب همگی درارتباط با یکدیگر هستند.و همگی دارای کلیه صفحات مورد تقاضا باید باشند.و همگی قدرت تصمیم گیری و انتخاب سرویس دهنده وب بعدی را دارند. به این ترتیب بار کاری در بین ایستگاهای کاری توزیع می گردد.

در اینجا دو مسله مطرح است که باید به آنها پرداخت، این دو مسله در ارتباط با انتقال اطلاعات بین صفحات یک متقاضی و همچنین انتقال اطلاعات سراسری در بین

متقاضیان متفاوت است. در ادامه به راه حل پیشنهادی خود برای این دو مسله می پردازیم.

اولا" چنانچه صفحه اول از روی یک سرویس دهنده انتخاب شده باشد، صفحه دوم بر روی هر سرویس دهنده دیگری که قرار داشته باشد، از طریق سرویس دهنده اولی اطلاعات به آن قابل ارسال خواهد بود.

ثانیا" می توان اطلاعات سراسری را در بین سرویس دهنده ها از طریق فیلدهای پنهان درون فرم برای صفحات بعدی ارسال نمود.

قابلیتهای برنامه نویسی با ASPو چگونگی کار با آن :

فناوری ASP یک روش برای ایجاد نبشته هایی(Script) است که در سمت کارگزار اجرا میشود و نتیجه ی آنها تولید پویا و ارسا ل صفحات ابر متن به سمت کاربر می باشد. ازاین فناوری می توانیم به منظور اجرای برنامه های کاربردی مبتنی بر وب استفاده کنیم،همچنین می توانیم صفحات ابر مبتنی دستورات نبشته و مولفه های Activexرا بایکدیگر ترکیب کنیم تا برنامه های کاربردی مبتنی بروب نیرومندی ایجاد شوند.

اگربا صفحات وب آشنا باشید در ادامه متوجه خواهید شد که نبشته های یک ASP راه آسان برای ایجاد صفحات پویای وب میباشد. به عنوان مثال ، اگرتا کنون خواسته باشید اطلاعات وارد شده توسط کاربر در یک فرم ابر متن را در سمت کارگزار استخراج و پردازش کنید و یا قابلیتهای نگاه ابزار (Browser) مشتری را تشخیص داده و از آن استفاده کنید، فناوری ASP امکانات کاملی را در این زمینه برای شما فراهم می آورد.

در گذشته اگر می خواستیم که اطلاعاتی را از فرمهای ابر متن جمع آوری کنیم، لازم بود با تسلط بر یک زبان برنامه نویسی، برنامه ایی نوشته و در سمت کارگزار قرار می دادیم که کار استخراج و پردازش اطلاعات مشتری را به عهده داشت. با استفاده ازِASP می توانیم با به کار گیری دستورات ساده ایی، اطلاعات یک فرم ابر متنی را جمع آوری کرده و آن ها را تحلیل کنیم و نیاز به یادگیری یک زبان سازی به صورت کامل نیست.

در صفحات ASP از زبانهای VBScript وJavaScript برای نبشته نویسی استفاده می شود. توجه به این نکته ضروری است که ASP یک تکنولوژی است و نه یک زبان، لذا اسکریپت صفحه ASP می تواند با هر زبانی نوشته شده باشد. و تنها کسی که کارگزار ابزار لازم برای اجرای این اسکریپت را داشته باشد.

تکنولوژی ASP :

ASP تکنولوژی جدید است که با میزبان وب مایکروسافت(IIS) می آید و برنامه نویسان را قادر می سازد تا برنامه هایی بنویسند که در کامپیوتر میزبان اجرا می شود. این یکی از قوی ترین ابزارهای ایجاد صفحات دینامیک وب است. اما ASP دقیقا" چگونه کار می کند؟یک صفحه وب شامل یک سند HTML و مقداری کد اسکریپ است که با پسوند ASP ذخیره میشود تا از صفحات معمولی HTMLمتمایز باشد. وقتی کاربر یک صفحه ASPرا باز می کند اتفاقات ذیل می افتد:

- میزبان تقاضایی برای صفحهASPدریافت می کند.
- میزبان صفحه را باز کرده و کد HTMLو اسکریپت آن را تحلیل می کند.

- بر اساس اسکریپت وHTML این صفحه یک صفحه جدیدHTMLایجاد می شود.
- صفحه جدید HTML به کامپیوتر مشتری فرستاده می شود و در کاوشگر آن نمایش داده میشود شکل زیراین فرایند را به تصویر کشیده است.

صفحهASP

.شکل سند ASPدر میزبان وب پردازش شده وبعد ازایجاد سند HTMLآنرا به مشتری میفرستد.

زبانهای اسکریپت نویسی Asp

توجه به این نکته مهم است که ASPیک تکنولوژی است نه یک زبان اسکریپت صفحه ASP می تواند با هر زبانی نوشته شده البته استفاده ازهر زبان ویژگیهای خاص خود را دارد بطور مثال درVbscriptنمی توان یک تابع را از خواندن آن صدا زد (صدا زدن

قبل از تعریف تابع نمی تواند صورت گیرد) ولـی در gavaScaipt ایـن کـار ممکـن است. البته از هر زبان مزایا و معایبی دارد که ما نمی خواهیم در اینجا به آن بپردازیم.

ساختار یک صفحه ASP :

صفحات ASPماند صفحات HTMLمبتی هستند. در این صفحات از همان برچسبهای HTML اسـتفاده مـی شـودو فقـط دارای تعـدادی سـاختار منطقـی هسـتند کـه د رHTMLوجود ندارد این ساختار با بر چسب <٪ ....٪> مشخص می شوند. سندی که به کاربر برگردانده می شود ترکیبی از HTML اولی که در صفحه ASP وجـود دارد و HTML ایجاد شده توسط اسکریپت ..

ASP دارای دو نوع مشخص کننده اسکریپت است که در بالا بـا یکـی از آنهـا یعنـی <٪....٪> آشنا شدیم. نوع دوم که از آن برای نمایش خروجـی عبـارات اسـتفاده مـی شـود چنین است:

< = % 000 %>

این برچسب می گوید که مقدار عبارت را درکاوش گر نمایش بده. مثلا" اگر total یـک متغییر Vbscrip باشد عبارت ذیل مقدار آن را در کاوش گر نشان خواهد داد.

< % = Total .. % >

اشیا میزبان فعال :

یک میزبان ASP اشیایی دارد که می توان در اسکریپت های نوشته شده از آنها استفاده کرد. این اشیا اطلاعاتی را درباره محیط میزبان در اختیار قرار می دهند، فرمها را مدیریت می کنند و داده ها را ذخیره می کنند برای آنکه بتوان برنامه های موثر ASP نوشت باید با این اشیا آشنا بود لذا در اینجا به آنها اشاره می شود.

Application

شی Application برای به اشتراک گذاشتن داده ها بین کاربران یک برنامه مورد استفاده قرار می گیرند. تمام فایلها وزیر دایر کتوری های برنامه جز این شی هستند و اطلاعات این شی در دسترس تمام صفحات و تمام کاربران قرار دارد.

توجه : شی Application اطلاعات نسبتا" محدودی را می تواند ذخیره کند برنامه های ASP نباید از این شی برای ذخیره کردن مقادیر حجیم داده استفاده کنند.

برای ایجاد متغییری از شی Application باید چنین عمل کرد :

Application,, Varname,, = Value

که در آن Varname نام متغییر و Value داده ای است که در آن ذخیره می کنند و برای خواندن این مقدار اگر هدف قرار دادن در متغییر My name باشد باید به طریقه زیر عمل کرد :

My name = Application (,, Varname,,)

اما اگر داده ها با متد Post ارسال شود از کلکسیون فرم استفاده خواهیم کرد :

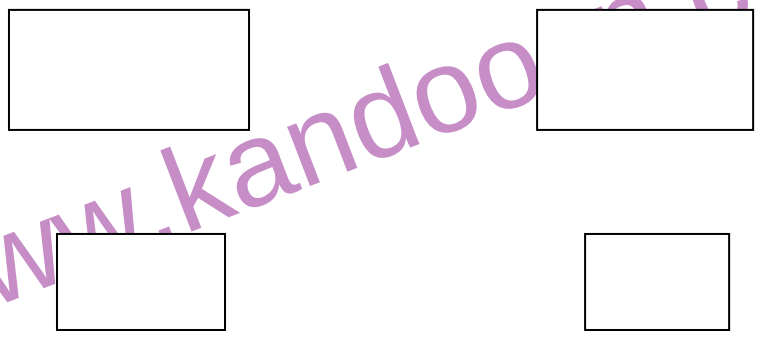
Var = Reouest.Form („ input Name„)

که در هر دو دستور input name نام عنصر ورودی HTML است کـه مـی خـواهیم داده های آن را بخوانیم.

اجرای برنامه سمت مشتری :

برای این کار باید از زبانهای خطی دستوری استفاده کرد. یکی از مزایای زبانهـای خطـی دستوری این است که آنها در کامپیوتر کار بر اجرا مـی شـودبه همـین دلیـل بـه آنهـا در سمت مشتری گفته می شودتا قبل از ظهور این زبانها هر تقاضایی به میزبان فرستاده مـی شد حتی ساده ترین کارها مثل تعیین صحت داده های ورودی باید در میزبان انجـام مـی شد و این مستلزم یک رفت و برگشت کامل بود این حالت نه تنها حجم کاری میزبـان را افزایش بلکه ترافیک شبکه را بالا می برد وباعث کندی کارها می شد.

زبان خطی دستوری (مانند Vbscript) اجازه می دهد تا انواع خاصـی از پـردازش داده ها در کامپیوتر مشتری انجام شود و احتیاجی به مکالمه بـا میزبـان وجـود نداشـته باشـد بدین ترتیب حجم کاری میزبان و ترافیک شـبکه کـاهش خواهـد یافـت. در شـکل زیـر مقایسه ای بین دو حالت را ملاحظه می کنید.



شکل زبانهای اسکریپت نویسی اجازه می دهند تا برخی از پردازشها به جای ارسال به کامپیوتر میزبان در همان کامپیوتر مشتری اجرا شوند.

زبانهای اسکریپت نویسی در داخل سندHTML به کار گرفته می شود دستورات این زبان با برچسب خاصی از سایر قسمتها مشخص می شوند. وقتی این صفحه به کاوشگری که از اسکریپت پشتیبانی می کند باز می شود کاوشگر اسکریپت را خوانده و آنرا اجرا می کند و کاوشگرهایی که از اسکریپت استفاده نمی کنند آنرا نادیده می گیرند.

برای جدا کردن اسکریپت از سایر قسمتها سند HTML از برچسب های </ Scirpt>.. </Scirpt> استفاده می شود در تمام زبانهای اسکریپت نویسی برچسب اول نوع زبان را هم مشخص می کند.

<SCRIPT LANGUAGE =,, VBSCRIPT ,,>

<SCRIPT >

استاندارد پذیرفته شده در اسکریپت نویسی قرار دادن کدهای اسکریپت نویسی در برچسبهای توضیح HTML یعنی ۰۰۰!> و <۰۰۰ است تا در کاوشگرهایی که از اسکریپت پشتیبانی نمی کنند متن کدها داده نشود.

<SCRIPT LANGUAGE = ,, VBSCRIPT,,>

<!..

..>

</SCRIPT>

تکنولوژی ASP به کاربر اجازه می دهد که هر دو فرم اجرا در سمت میزبان (به طور نا محدود) و اجرا در سمت مشتری (به طور محدود)را به طور توام استفاده کند.

ارتباط با بانک اطلاعاتی:

با آن که در تکنولوژی ASPنیز برای اتصال از پایگاه داده مدل شی ADOاستفاده می شود ولی تفاوتهایی در این مدل با مدل مشابه در اکتیواکس وجود دارد که در ذیل بیان خواهد شد. در مدل ADO ارائه شده برای ASPجهت برقراری ارتباط با بانک اطلاعاتی باید از شی Connectionاستفاده کرد. این شی خواص متعددی دارد که ما تنها به string Connection از میان آنها نیاز داریم که این خاصیت همان طور که از نام آن پیداست جزئیات اتصال به پایگاه داده(تنظیمات خواندن/ نوشتن/ اشتراک) از اطلاعاتی هستند که توسط خاصیت Connection string مشخص می شود. مانند دستورات ذیل :

```
<% cs= ,,provider = Microsoft.jet.OLEDB.3.51,
cs = cs &,, Data source = &Server.Map path (          )&,, . ,,
cs = cs&Mode = Reade =Read/Write/share Deny None % >
```

که در نهایت رشته اتصال را مشخص می کند به طوری که نوع ارائه کننده (Provider) مکان قرار گرفتن پایگاه داده(Data Source) و حالت اتصال به پایگاه

داده(Mode)در آن معین شده است. شی Connection دارای سه متد است که عبارتند از :

متد Open :ارتباط بین برنامه و فایل پایگاه داده ها را بر قرار می کند شکل کلی این متد چنین است :

Connection . Open Connection steing, User ID , Password

تمام آرگونهای این متد اختیاری است چون حتی رشته اتصال را می توان در خاصی Connection string شی Connection قرار داده و متد open.را بدون هیچ آرماگونی اجرا کرد. اگر متد open با شکست مواجه شود یک خطا به کلکسیون Errorشی connection اضافه می شود.

متد Error شی Connection :برای ارسال یک فرمان،به پایگاه داده استفاده می شود.

Connection. Execute command

که در آن Commandفرمانی است که باید اجرا شود نتیجه اجرای فرمانها همیشه به صورت یک Recordsetبر گردانده می شود مانند رابطه زیر :

Din rs

Set rs = cn.Execute („ Select *from)

بعد از اجرای این دستور رگوردست حاوی تمام رکوردهای جدول نام برده خواهد بود.

متد :Close

متد دیگر شی connection یعنی close ارتباط با پایگاه داده را قطع کرده و منابع سیستم را آزاد می کند اجرای این متد شی connection را از بین نمی برد برای این کار باید به این شی مقدار Nothing داده شود.

رکوردست :

هر گونه کاری روی داده ها در ADO از طریق شی رکوردست صورت می گیرد هر رکوردست یک اشاره گر رکورد (record pointer) دارد که به مکان غعلی آن اشاره می کند اگر این اشاره گر به هیچ رکوردی اشاره نکند می تواند در ابتدا یا انتهای دارای متدهای زیر می باشد:

Move NumRecs :اشاره گر رکورد را به اندازه NUMRecs جابجا می کند اعداد منفی اشاره گر را به سمت ابتدا و اعداد مثبت اشاره گر را به سمت انتهای جدول منتقل می کند.

Move provider :اشاره گر رکورد را به آخرین رکورد در رکوردست منقل می کند.

Move Next : اشاره گر رکورد را به رکورد بعدی منتقل می کند.

Move p tevious : اشاره گر رکورد ره به رکورد قبلی منتقل می کند.

نکته : در ASP مدل ADO شامل دستورات find برای یافتن رکورد مورد نظر نمی باشد لذا بدین منظور باید خودمان این تابع را بنویسیم.

برای بازیابی داده های فیلد مورد نظر از رکوردست به اشکال زیر می توان عمل کرد :

الف:

<% = Recordset.fields(Index).Value % >

<% = Recordset.fields(,, Name ,,) Value % >

< % = Recordset (,, Name ,,)

که در مورد ب و ج Name اسم فیلد در بانک اطلاعاتی و در رابطه الف به جـای اسـم از اندیس که باید یک شماره باشد استفاده می شود که این شماره فیلد در بانک اطلاعاتی است.