مقدمه

برای ساخت صفحات وب از برنامه HTML استفاده می شود که در زیر به شرح آن میپردازیم.

تاریخچه HTML:

در سال ۱۹۹۲ در دانشگاه مینه سوتا سیستمی به نام web به وجود آمد که دارای دو ویژگی خاص بود.

۱. Graphic, Multi Media

۲. Hyper Text (فقط کلیک کردن و در سایت حرکت کردن)

صفحات این سیستم توسط برنامه ای به نام HTML ساخته شد.

همچنین برای رد و بدل کردن اطلاعات:

HTTP (Hyper Text Transfer Protocol)

HTML Web page

HTML یک text عادی و در حقیقت زبانی برای مارک کردن فایلهای text به یکدیگر می‌باشد که آن را با TAG مشخص کرده و به صورت < tag name > می نویسند.

1- فرمت کلی یک فایل HTML:

یک فایل HTML از دو بخش Body و Head ،تشکیل می شود. شکل ساده یک فایل HTML به صورت زیر است.

< HTML >

< HEAD >

< TITLE > This is the title< /TITLE >

< HEAD/ >

This is the body

< BODY/ >

< HTML/ >

نمایش صفحه ای که دارای HTML بالاست...

در بالای صفحه حاصل،متن نوشته شده در قسمت Title و در بدنه

اصلی صفحه متن ما نوشته شده است.

1-1. تگ HEAD

در برچسب HEAD از برچسب به نامهای TITLE و BASE و META استفاده می‌شود.

< HEAD >

< TITLE > < /TITLE >

< META > < /META >

< BASE >

< HEAD/ >

1-1-1. TITLE:

برای تعیین لقب صفحه (چیزی که در قسمت Status Bar دیده می شود

1-1-2. META:

_ برای تعیین نام و منبعی که برنامه توسط آن نوشته شده.

_ بهنگام کردن صفحات web توسط این برچسب انجام می شود.

_ انتقال به یک صفحه دیگر web در زمان معیین.

مثال برای حالت اول:

در این حالت برای وارد کردن آدرس web خودمان به موتورهای جستجو در web (مثل yahoo و google و ...) از META استفاده می کنیم:

< META name="keyword"{اجباری} content="Hedayat, students,zahiri,yaghoubi,schoolnet") >

< META name="description"{اجباری} content="This is Hedayat high school" >

مثال برای حالت دوم و سوم:

< META name="vali "{دلخواه} http_equiv="refresh" content=" زمان بر "حسب ثانیه" >

با این برچسب صفحه web بعد از ۱ دقیقه بهنگام (refresh) خواهد شد.

در مثال بالا اگر در قسمت content به صورت زیر عمل کنیم صفحه web بعد از ۶۰ ثانیه به http://www.schoolnet.ir خواهد رفت.

:BASE .3-1-1

برای مشخص کردن مبدا آدرس دهی از صفحات web می باشد.

< BASE href="آدرس" >

< BASE href="http://www.schoolnet.ir/~zahiri/index.htm" >

نکته مهم:

در برنامه نویسی HTML برچسب ها به دو صورت با پایان و بی پایان نوشته می شوند.

ا لف -< TAG >.....< TAG/ > با پایان

ب- < TAG > بی پایان

2-1. تگ BODY

قسمت دوم یک فایل HTML را Body تشکیل می دهد که دارای Attributeهای زیر می‌باشد.

BODY bgcolor = "رنگ پس زمینه صفحه"

background = "آدرس عکسی که به عنوان پس زمینه در صفحه وب قرار می گیرد. "

topmargin = "یک فضای خالی بالای صفحه بر حسب پیکسل ایجاد ما کند"

leftmargin = "یک فضای خالی سمت چپ صفحه بر حسب پیکسل ایجاد ما کند"

text = "color" رنگ متن را مشخص می کند

link = "color"

alink = "color" (active link)

vlink = "color" (visited link)

نکته مهم:

در برنامه HTML و در نوشتن تگها بزرگ و یا کوچک نوشتن حروف هیچ تاثیری ندارد.

- **آشنایی با برخی تگها:**

Font -2:

با این برچسب می توانیم مشخصات متن را به دلخواه خود درآوریم و فرمت کلی آن به‌صورت زیر است.

< FONT/ >..... <FONT >

این تگ دارای Attributeهای زیر می باشد:

۱ .colorرنگ متن

۲ .size: اندازه متن

۳ .face: نوع متن

مثال: می خواهیم کلمه Schoolnet را با فونت نازنین و با اندازه normal و رنگ آبی بنویسیم.

< HTML >

< HEAD >

< HEAD / >

< FONT/ > Schoolnet < ″FONT size = "3" color ="blu e >

< BODY/ >

< HTML/ >

نمایش صفحه ای که دارای HTML بالاست...

متن ما به رنگ آبی درآمده و اندازه آن نیز ۳ شده است.

نکته: اگر بخواهیم اندازه را نسبی مشخص کنیم یعنی نسبت به آنچه که قبلا بوده به صورت زیر عمل می کنیم:

Size = +2

BOLD :

اول و آخر متن مورد نظر قرار گرفته و آنرا Bold می کند.

< B/ > … text < B >

ITALIC:

اول و آخر متن مورد نظر قرار گرفته و آنرا Italic می کند.

< I/ > … text < I >

UNDERLINE:

اول و آخر متن مورد نظر قرار گرفته و آنرا Underline می کند.

< U/ > … text < U >

ANCHOR:

در HTML بوسیله تگ < a > ... < /a > می توانیم یک متن یا عکس را به صفحه‌ای دیگر پیوند دهیم (Hyper link).مهمترین Attribute در این تگ، href می باشد.

فرمت کلی این تگ به صورت زیر است.

< a/ > image/ text < ”a href = “ URL >

مثال: در جمله click here to go to zahiri home page کلمه zahiri را به آدرس index .html لینک کنیم:

< BODY >

< p >

click here to go to

< a/ > schoolnet < ″a href = "http: //www.schoolnet.ir >

home page

< p/ >

< BODY/ >

نمایش صفحه ای که دارای HTML بالاست...

کلمه مورد نظر به آدرسی که می خواهیم ، لینک شده است

تگهایی که به وسیله آنها متن را فرم بندی می کنیم:

PARAGRAPH -3

< P/ > ...: در این تگ همه موضوعات آن در یک خط نوشته شده یا یک پاراگراف ایجاد می‌شود که در صورت بوجود آمدن پاراگراف در زیر خط اول، خط دوم را با فاصله زیاد می‌نویسد. برای حل این مشکل از تگ < B R > استفاده می کنیم.

تگ < P > دارای یک Attribute است:

< ″A align = " left /center/right >

BREAK -4

< BR >: این تگ از تگهایی است که پایان ندارد و آن را هر کجا که قرار دهیم کلمه بعدی را در یک خط پائین تر ولی با کمتر می نویسد.

BREAKN -5

… < /NOBR > : اگر بخواهیم در آخر خط شکستگی نداشته باشیم بین دو بخشی که شکسته می شود از این تگ استفاده می کنیم.

مثال: < NOBR/ >a2

a, 2 را هرگز از هم جدا نمی کند.

HEADING -6

در HTML دارای شش نوع HEADING هستیم.

بزرگترین < H1/ > … <H1 >

< H2/ > … < H2 >

.

.

کوچکترین < H6/ > … < H6 > .

خود این تگها خاصیت راست چین و چپ چین و یا وسط چین شدن را هم دارند کـه بـرای فعال کردن آن از روش زیر استفاده می کنیم.

مثال: < H2/ > vali < "H2 align="center >

HR -7

تگ < HR > برای ما تک خط افقی سه بعدی ایجاد مـی کنـد و دارای Attributeهـای زیر می باشد.

HR align="left/center/right" width ="طول خط بر حسب پیکسل یا درصد"

size="ضخامت خط بر حسب پیکسل"

noshade: با نوشتن این کلمه خط سه بعدی نمی شود .

color="رنگ خط"

PRE -8

در HTML هر چیزی را که بین تگ < PRE > ... < PRE/ > به هر صورتی که بنویسیم با همان شکل در صفحه وب نشان می دهد.

مثال:

Vali < PRE >

… Ali

Reza a b cd

< PRE / >

همانطور که می بینید دقیقا همان طور که متن در HTML، نوشته شده اسـت در صـفحه اصلی دیده می شود.

DIV -9

برای ما یک بلوک در متن ایجاد می کند.

مثال < DIV/ > … … < ″style="color:red

استفاده دیگر برای تعیین Direction می باشد که سمت نوشتن را از راست به چپ یا چپ به راست می کند.

< DIV/ > … … < ″dir="rtl/ltr DIV >

rtl = right to left

right ltr = left to

از Attributeهای این تگ خاصیت align می باشد:

< DIV/ > … … < ”align=”left/center/right DIV >

در قدیم از تگهای دیگری نیز برای ویرایش متن استفاده می کردند که اکنون بیشتر آنها منسوخ شده اند واستفاده ای ندارند.

در زیر به برخی از آنها اشاره می کنیم:

< e m > … < e m/ >: دقیقا کار تگ < I > … [i /] را می کند ولی نشان مـی دهـد که از نظر منطقی تاکیدی روی متن می باشد.

< s trong > … < s trong / >: دقیقا کار تگ < B > … [b /] را انجام می دهد.

< c ite > … < c ite / >: برای نوشتن اسم مقاله یا کتاب (آنرا italic می کند)

< c ode > … < c ode / >: برای نوشتن قطعه برنامه از آن استفاده مـی شـود. (بـا خـط Scape Mono)

< k bd > … < k bd / >: ورودی های برنامه را با این تگ می نوشته اند.(M.S)

< s amp > … < s amp / >: برای نوشتن مثال از این تگ استفاده می شده.(M.S)

< v ar > … < v ar / >:متغییرها را Italic می کند.

< t t > … <t t / >: متن را Mono Scape می کند.(یک نوع Font است.)

< s trike > … < s trike / >: یک خط وسط متن داخل خود میکشد.

< S TRIKE > a bc < S TRIKE />

که نتیجه کار چنین است.

< b lockquote > … < b lockquote / >:

< b ig > … < b ig / >: متن که بین آن باشد یک فونت درشتتر می نویسد.

< s mall > … < s mall / >: متن که بین آن باشد یک فونت کوچکتر می نویسد.

برای درست کردن توان و اندیس دو تگ داریم.

مثال: برای نوشتن از دو تگ زیر استفاده می کنیم.

< S UB / > ۱ < S UB > a

< S UP / > ۲۰ < S UP > a

… < / blink >: این تگ فقط در NetScape کار می کند. متن یا عکس که داخل این تگ قرار می گیرد در صفحه چشمک می زند.

فصل اول

پروژه یا سایت اختصاصی بیمارستان مهر با زبان ASP.NET نوشته شده است که حال زمان آن رسیده تا به شرح این زبان بپردازیم :

1- چرا به دات نت احتیاج داریم؟

به طور معمول نسل های جدید زبان های برنامه نویسی به این دلیل متولد می شوند که زبان های قدیمی تر دارای امکانات محدود بودند و یا قدرت استفاده از تکنولوژی های فعلی را به صورت مطلوب و ساده ندارند.

مهمترین نیازی که به عنوان آخرین تکنولوژی وجود دارد، برنامه نویسی در محیط اینترنت است. اینترنت در مدت تقریبا ۸ سال جای خود را به عنوان یکی از مهمترین وسایل ارتباطی برای کارهای روزمره و تجارت باز کرده است.

سیستم های برنامه نویسی قدیمی تر امکان برنامه نویسی برای اینترنت را فراهم کرده بودند ما هر کدام دارای اشکالات بزرگی هستند، برای مثال تکنولوژی COM اولین بار در ویندوز به کار گرفته شد. در سال ۱۹۷۰ نیز سیستم هایی برای Unix نوشته شده بودند، جاوا نیز در اصل برای ابزارهای الکترونیکی بود و نه برای اینترنت.

سپس برای اولین بار یک سیستم جامع برای برنامه نویسی تحت اینترنت ایجادشد. این سیستم -NET. از مراحل سطح پایین که به زبان ماشین می باشد تا بالاترین سطح که برنامه نویسی ویژوال آن می باشد برای استفاده در اینترنت طراحی شده است. البتهNET. فقط برای اینترنت نیست و با استفاده از آن می توان برنامه های کامل تحت Client نیز ایجاد کرد، اما بزرگترین مزیت آن دربرابر سیستم های دیگر امکانات اینترنت آن است.

برای اینکه مزایای استفاده ازNET. را بهتر متوجه بشویم بهتر است در ابتدا معایب سیستم‌های پیشین را ذکر کنیم.

شرکت مایکروسافت تا قبل از سال ۱۹۹۵ به برنامه نویسی در محیط های Client و Server می پرداخت، اما از آن سال به بعد توجه بیشتری به مساله برنامه نویسی در اینترنت کرد.

مایکروسافت COM و +COM را ایجاد کرد و آنها را در ویژوال استودیوی ۶ به کار گرفت. در سال ۱۹۹۹ حدود ۵۰ در صد از بزرگترین سایتهای تجارت الکترونیکی از محصولات مایکروسافت استفاده می کردند.

اما هنوز هم مشکلات بزرگی در سیستم های مایکروسافت وجود داشت که یکی از آنها دشواری نوشتن برنامه در اینترنت با محصولات مایکروسافت بود.

شرکت مایکروسافت برای راحتی کار برنامه نویس ها ASP یا Active Server Page را ایجاد کرد. با اینکه این یک قدم بزرگ بود و کارها را بسیار ساده کرد ولی هنوز از برنامه نویسی شی گرا پشتیبانی نمی کرد. همچنین در ویژوال استودیوی ۶ قسمتی برای Internet Application ایجاد شده بود و در آنها امکان ساختن Web Class وجود داشت ولی هیچ وقت به عنوان یک ابزار کار آمد برای برنامه نویسی وب درنظر گرفته نشد.

2- مدل برنامه نویسی DNA

مایکروسافت یک مدل برنامه نویسی به نام Application Distributed interNet دارد که بر پایه برنامه نویسی n-tier و COM بنا نهاده شده است. مدل DNA از سه بخش اساسی تشکیل شده است.

بخش اول به نام Presentation tire معروف است. در ایـن بخـش رابـط تـصویری کـاربر وجود دارد و خود نیز به دو نوع Internet Browser و Win 32 GUI تقسیم می شود کـه هر کدام مشکلات خاص خود را دارند. در مدلی که از GUI Win32 یا همـان نـرم افزارهـای معمولی استفاده می شود دو مشکل بزرگ وجود دارد ؛ دشواری بروز رسانی نرم افزار و دیگـری DLL Hell که در ادامه توضیح داده خواهد شد.

در نوع دوم مشکلاتی از قبیل نبود امکانات برنامه نویسی کافی در محیط مرورگر، نبود رابط قوی با کاربر، نبودن مرورگر های یکسان و... وجود دارد.

همچنین همیشه یک اتصال به اینترنت یا اینترانت لازم است. در این نوع از برنامـه نویـسی می توان از Java Applet ها یا ActiveX استفاده کرد ولی مرورگر بایـد امکـان اسـتفاده از آن را داشته باشد، مخصوصاً هنگام استفاده از ActiveX که باید فقط از IE استفاده کرد.

بخش دوم که Middle tier نام دارد، مکانی اسـت کـه اطلاعـات و قـوانین تجـاری در آن وجود دارد. منظور از قوانین ، متد ها و اجزائی هستند که اعمال کاربران را کنتـرل مـی کننـد. مهمترین و آسان ترین زبان برای نوشتن این اجزا از DNA ویژوال بیسیک است.

برنامه نویسی کـه بخواهـد در ایـن رده برنامـه بنویـسد بایـد آشـنایی کـاملی بـا COM و پروتکل‌های رایج داشته، همچنین باید مهـارت کـافی در اسـتفاده از ADO و ADSI داشـته باشد. مشخص است که یک اشتباه در این لایه باعث بروز خطا و نقص در کل سیستم می شود.

بخش سوم یا Data tier مکانی است که اطلاعات سازمان در آن ذخیره می شود. معمـولاً در این قسمت از بانکهای پیـشرفته رابطـه ای ماننـد SQL Server و Oracle اسـتفاده مـی کنند.

3- محدودیت های COM

همانطور که دیدید مهمترین قسمت در DNA همان COM است که در جای جای آن استفاده می شود. در اینجا برخی معایب COM ذکر می شود : (در ابتدای متن ذکر شد که برای درک نیاز به.NET باید ابتدا معایب سیستم های قدیمی را بشناسیم)

DLL Hell: اگر کوچکترین تغییری در یک COM ایجاد شود، دیگر برنامه هایی که از ورژن قبلی استفاده می کردند قادر به فعال ساختن نسخه جدید نیستند. هنگامی که در ویندوز، یک COM نصب شود برایش در رجیستری یک GUID ثبت می شود که اطلاعات آن COM را در خود ذخیره می کند.

اگر یک برنامه از نسخه اول یک COM استفاده کند و بعد از مدتی شما تغییراتی در نسخه اول بدهید و بخواهید آن را دوباره در سیستم نصب کنید ویندوز به شما پیغام خطا می دهد چون ورژن آن تکراری است، اگر هم آن را به ورژن دوم ارتقا دهید نرم افزار قبلی هنوز به دنبال نسخه اول می گردد. این امر باعث می شود که شما مجبور شوید یکبار دیگر کل برنامه را کامپایل کرده و در کامپیوترتان نصب کنید.

کمبود در وراثت: در نسخه های COM که در حال حاضرهستند چیزی به نام وراثتی که در C++ وجود دارد نمی باشد، بلکه وراثت تنها در واسط یک COM می باشد، استفاده از آن هم چندان کمکی به برنامه نویسی نمی کند.

برخی محدودیت های برنامه نویسی اینترنتی در مدل DNA :

ا لف: وجود دو محیط برنامه نویسی برای اینترنت و Client

نقصان در نوشتن برنامه هایی با رابط گرافیکی خوب که در اینترنت کار می کردند کاملاً مشهود است، نمونه بارز آن اختلاف در برنامه نویسی در ویژوال بیسیک و ASP است. ویژوال‌بیسیک با رابط گرافیکی کاملاً سطح بالا و ASP تقریباً رابط گرافیکی ندارد.

همین امر باعث می شد که یک برنامه نویس مجبور باشد طیف وسیعی از تکنیک ها و زبان‌ها را فرابگیرد تا بتواند برنامه ساده ای در اینترنت بنویسد.

ب: نبودن حالت های ذخیره اطلاعات رابط گرافیکی در صفحه های اینترنتی

نمونه این حالت زمانی است که در یک textbox متنی وجود داشته باشد. در برنامه های Win32 GUI متن داخل textbox تا زمانی که کاربر یا برنامه آن را تغییر نداده بر جای خود وجود دارد. اما در محیط اینترنت و نوع ASP با هر بار refresh کردن صفحه کل اطلاعات ازبین می رود. البته این مشکل با استفاده از شئ های Request و Response تقریباً قابل حل است ولی احتیاج به برنامه نویسی برای هر تکه از صفحه ASP دارد.

ج: نداشتن Event Handler در محیط برنامه نویسی اینترنت

یکی از مهمترین ابزاری که در برنامه نویسی Win32 GUI وجود دارد استفاده از Event ها است. با تکنولوژی که در حال حاضر وجود دارد تنها راه رسیدن به این مهم استفاده از ActiveX است که به علت مسایل امنیتی در بیش از ۹۵ در صد مواقع توسط کاربر استفاده از آن رد می شود.

4- معایب استفاده از API

API ها توابعی هستند که از ویندوز نسخه ۱ تا امروز در برنامه نویسی کاربرد داشته و دارند.مهمترین کاری که این توابع انجام می دهند انجام کارهای سخت و سطح پایین سیستمی است که احتیاج به برنامه نویسی زیادی دارند و یا حتی امکان ایجاد آن با زبان هایی مثل ویژوال بیسیک نیست. اما هر API از هر نسخه ویندوز تا نسخه دیگر آن می تواند دچار تغییرات بشود.

برای مثال برنامه ای که در ویندوز ۹۸ نوشته شده باشد می تواند در ویندوز ۹۵ اجرا نشود. همچنین هم اکنون ابزارهای جدیدی به بازار آمده است که برای آنها نیز می توان برنامه‌نویسی کرد، مانند تلفن های سیار، کیوسک تلفن، دستگاه های کامپیوتری جیبی و غیره.

در این نوع دستگاه ها دیگر ویندوز به مفهومی که در حال حاضر وجود دارد قابل اجرا نیست و در نتیجه API هم وجود ندارد. لازم به ذکر است که ویندوز CE برای دستگاه های مذکور می باشد ولی قابلیت های آن با ویندوزهای دیگر تفاوت زیادی دارد.

5- آشنائی با ASP.NET

NET نسل بعدی Active Server Pages یا ASP است که توسط شرکت میکروسافت ارائه شده است. این محصول توسط میکروسافت بعنوان شاخص اصلی فناوری در ساخت سایتهای وب در نظر گرفته شده است.

با استفاده از ASP.NET می توان هم اینترانت 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 کارائی و قابلیت آن می باشد.

در زیر برخی ویژگیهای ASP.NET را بررسی می کنیم:

• صفحات ASP.NET کامپایل می شوند.

هنگامی که یک صفحه ASP.NET برای اولین بارتوسط یک مراجعه کننده به سایت فراخوانی می شود، آن صفحه ابتدا کامپایل شده و بر روی سرور نگهداشته 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 DataGrid می توان به آسانی داده های موجود در یک بانک اطلاعاتی را تحت وب نمایش داد.

• مجموعهASP.NET عضوی از بدنه .NET است.

بدنه .NET دارای بیش از ۴۵۰۰ کلاس آماده جهت استفاده در ASP.NET است. این کلاس ها تقریبا هر نیازی را در برنامه نویسی برآورده می کنند. بعنوان مثال از این کلاس ها

می‌توان جهت تولید تصاویر بر حسب تقاضا، به رمز درآوردن یک فایل و یا ارسال یک نامه استفاده کرد.

6- مقایسه ASP.NET و ASP کلاسیک

ASP.NET نسل بعدی ASP یا ASP کلاسیک است. اما این یک پیشرفت تکاملی است بطوریکه این دو فناوری تقریبا از یکدیگر متفاوتند.

صفحات ASP با زبان های دستورالعمل نویسی مانند VBScript یا JScript ایجاد می‌شوند اما در ASP.NET ما یک فرایند کامل برنامه نویسی با زبانهای Visual Basic یا C# (سی-شارپ تلفظ شود) داریم. همچنین در ASP کلاسیک تنها پنج کلاس استاندارد (Application Session, Server ,Request, Response) وجود دارد حال آنکه در ASP.NET می توان 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 می توان یک صفحه بسیار پیچیده HTML بدست آورد که ساخت آن در ASP کلاسیک ممکن است نیاز به چند روز کار داشته باشد.

7- زبانهای برنامه نویسی در ASP.NET

شما در ASP.NET می توانید از هر زبان برنامه نویسی که با بدنه NET. سازگار باشد استفاده کنید. این زبانها عبارتند از Visual Basic.NET و C# و JScript.NET . 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 می دانید هم اکنون می توانید 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 شده است. بعنوان مثال اگر مایل باشید حتی می توانید از PERL و COBOL هم در ASP.NET استفاده کنید.

8-ابزارهای ASP.NET

سالهاست که برنامه نویسان ویژوال بیسیک جهت ساخت فرم های خود از ابزارهای ویژوال بیسیک مانند TextBox و ListBox استفاده کرده اند. در ASP.NET هم شما می توانید از ابزارهای فراوان موجود در آن برای ساخت فرم ها و صفحات خود استفاده نمائید. در ASP.NET چهار دسته عمده از ابزارها موجود است:

• ابزارهای اصلی مانند TextBox، RadioButton، ListBox و Button.

• ابزارهای اعتباری برای حصول اطمینان از ورود و تائید صحت اطلاعات ورودی فرم ها.

• ابزارهای داده ای برای ارتباط با بانک اطلاعاتی و دستکاری داده.

• ابزارهای پیشرفته جهت نمایش عناصر پیچیده در واسط کاربر مانند تقویم و آگهی های تبلیغاتی.

با استفاده از Studio.NET Visual شما براحتی می توانید با چیدن تصویری این ابزارها بر روی فرم مورد نظر، صفحه دلخواه خود را بسازید. در صورت تمایل حتی می توانید در یک ویرایشگر ساده متن مانند Notepad برنامه مورد نظر را نوشته و از این ابزارها استفاده کنید.

9- دریافت ASP.NET

جهت شروع برنامه نویسی در ASP.NET تنها کافی است که مجموعه ASP.NET را به‌همراه بدنه NET. از سایت میکروسافت دریافت کنید.

10- دریافت .NET Framework

ASP.NET بـــا سیـــستم عامـــل هـــای Windows 2000 (نـــسخه Server و Professional) و Windows XP کاملا سازگار است.

11-Namespace چیست؟

یک نکته مهم که در زمان استفاده از .NET Framework باید بـه آن توجـه داشـت آن است که فضانام (namespace یا نامکده) ها در ساختمان برنامه کاربردی قرار دارند.

فضانام یک طرح نامگذاری منطقی برای گروه بندی کلاس های مرتبط است. این طرح مانع از آن می شود تا کلاس هایی که برای متدها و خصوصیات از یک شناسه یکسان استفاده می کنند تداخل داشته باشند.

مثلا .NET Framework برای گروه بندی تایپ ها به مقولـه هـای منطقـی عملکـرد، از قبیل چارچوب برنامه کاربردی ASP.NET، از یک طرح نامگذاری سلسله مراتبی استفاده می کند.

ابزارهای طراحی از فضانام ها با هدف تسهیل مرور و ارجاع تایپ ها در برنامه بهـره بـرداری می کنند. مثلا فرض کنید در حال نوشتن کد زیر هستید:

```
Public Class NewClass
[Procedures and Functions]
End Class
Public Class NewClass
[Procedures and Functions]
End Class
```

این کد به خطا منجر می شود چون کامپایلر راهی برای تشخیص کلاس ها از یکدیگر ندارد. برای غلبه بر این مشکل می توان از یک فضانام استفاده کرد که اجازه می دهد دو کلاس همنام در صفحه با هم وجود داشته باشند. قطعه برنامه زیر تعریف این دو کلاس در فضانام های منحصر بفرد را نشان می دهد:

```
Namespace One
Public Class NewClass
[Procedures and Functions]
End Class
End Namespace
Namespace Two
Public Class NewClass
[Procedures and Functions]
End Class
End Namespace
```

در این کد برخوردی بین دو کلاس با نام NewClass وجود ندارد چون هر کدام در یک فضانام جداگانه قرار داده شده است.

کلاس اول را می توان با استفاده از ترکیب One.NewClass صدا زد، حال آنکه کلاس دوم را می توان با استفاده از ترکیب Two.NewClass صدا زد.

شما می توانید در فضانام های خود از یک ساختار سلسله مراتبی استفاده کنید.

قرار دادن اشیاء مشابه تحت زیرعنوانها در یک فضانام مشترک تشخیص هدف فضانام را آسانتر می کند، و در عین حال باعث می شود برنامه به مراتب شئ گراتر شود.

برای توضیح فضانام می توان ساختار فایل و دایرکتوری (کشو، فولدر) در یک کامپیوتر را در نظر گرفت. در این مثال کلاس ها به مثابه فایل ها و فضانام ها مانند دایرکتوری ها هستند. بدیهی است همانگونه که می توانیم دایرکتوریهای تو در تو داشته باشیم، فضانام ها هم در حالیکه کلاس ها را در خود جای داده اند می توانند بصورت تو در تو باشند.

فضانام ها در ساخت برنامه های کاربردی ASP.NET نقش مهمی ایفا می کنند. خوشبختانه لازم نیست برای همه اشیایی که می توانند به وسیله صفحات ASP.NET مورد استفاده قرار بگیرند سیستم طبقه بندی فضانام ایجاد کنید.

مایکروسافت این مساله را برای شما حل کرده است. دو فضانام ریشه، و فضانامهای فرزند آنها را می توان وارد صفحات ASP.NET خود کرد. اولی System نامیده می شود، و دومی Microsoft نام دارد. این فضانامها با جزئیات بیشتر در ادامه مورد بحث قرار گرفته اند.

12- فضانام System

فضانام System فضانام اصلی برای ساخت ASP.NET و همه برنامه های کاربردی دیگر مبتنی بر .NET Framework است. هر چیزی که در برنامه کاربردی شما قابل انجـام باشـد از طریق فضانام System کنترل می شود.

به عنوان مثال کنترل آرایه، عملیات ریاضی، و تبدیل نوع داده ها از طریق فضانام System و فضانامهای فرزند آن اداره می شوند. ۹ فضانام پیش فرض (فـضانام System و ۸ فرزنـد 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

12-1. System.ComponentModel.Design:

دربرگیرنده کلاس هایی است که می توان از آنها برای طراحی پشتیبانی سفارشی اجزا و زمان طراحی و دسترسی به سرویس های تامین شده توسط معماری .NET Framework استفاده کرد.

12-2. System.Data:

امکان دسترسی به کلاس ها و رابطهایی را فراهم می کند که معماری ADO.NET را برای دسترسی به داده های عمومی تشکیل می دهند.

12-3. System.Drawing:

دربرگیرنده کلاس ها و رابطهایی است که عملکرد گرافیکی اولیه را تامین می کنند. فضانام System.Drawing نیز از طریق فضانام System.Drawing.Drawing2D و System.Drawing.Imaging عملکرد پیشرفته تری فراهم می کند.

System.Web .4-12:

کلاس ها و رابطهایی تامین می کند که ارتباط مرورگر/سرویس دهنده را امکان پـذیر مـی کنند. این فضانام دربرگیرنده کلاس HTTPRequest (فراهم کننده اطلاعات وسـیعی دربـاره درخواست HTTP جاری)، کلاس HTTPResponse (فـراهم آورنـده امکـان دسترسـی بـه فرآیندها و یوتیلیتی های سمت سرویس دهنده) است.

System.Web.SessionState .5-12:

فراهم کننده کلاس ها و متدهایی برای مدیریت وضعیت جلسات کاری می باشد.

System.Web.UI .6-12:

فراهم کننده کلاس ها و رابطهایی برای رابـط واسـط کـاربر برنامـه کـاربردی ASP.NET است که موجب می شوند برنامه کاربردی با سطوح مختلف صفحه، ارتباط برقرار کند.

کلاس اصلی این فضانام، کلاس Page می باشد که دربرگیرنده همه خصوصیتها، متـدها، و سازنده های صفحه است. اشیاء اصلی Active Server Page زیـر خـصوصیتهایی در کـلاس Page هستند: Application، Response، Request،Server و Session.

12-7. System.Web.UI.HTMLControls:

کلاس هایی برای عناصر HTML استاندارد، شامل فرم ها، کنترل های ورودی، آنکور، جداول، قسمتهای متنی، و غیره فراهم می کند. این کنترلها همانند تگهای عادی HTML هستند با این تفاوت که داری دو صفت runat="server" و id="controlname" می باشند.

12-8. System.Web.UI.HTMLControls:

برای کنترلهای سرویس دهنده ای که شبیه کنترلهای HTML هستند ولی انعطاف پذیری بیشتر و عملکرد پیچیده تری دارند کلاس هایی را تامین می کند.

برخی فضانام های مهم و پرکاربرد دیگر به شرح زیر می باشند.

• System.IO: دربرگیرنده رابطها و کلاس هایی است که امکان خواندن و نوشتن همگام و غیرهمگام فایل ها و جریانهای داده را فراهم می کنند.

• System.Data.OleDb: امکان دسترسی به کلاس ها و رابطهای مخصوص دسترسی به یک منبع داده از طریق ADO را فراهم می کند.

• System.Data.SqlClient: امکان دسترسی به کلاس ها و رابطهای مخصوص دسترسی به داده های خاص Microsoft SQL Server از طریق ADO را فراهم می کند.

• System.Web.Security: امکان دسترسی به کلاسها و رابطهای مخصوص امنیت برنامه کاربردی ASP.NET را فراهم می کند. دستیابی به رمزنگاری، مجوزها، و تنظیمات خط‌مشی برنامه کاربردی در این فضانام قرار می گیرند.

• System.XML: امکان دسترسی به کلاسها و رابطهای مخصوص پردازش اسناد XML را فراهم می کند.

13- فضانام Microsoft

علاوه بر فضانام System که در چارچوب .NET یافت می شود، مایکروسافت چند فضانام 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 فضانام کاری نداشته باشید.

13-1. Microsoft.VisualBasic:

این فضانام محتوی CLR یا زمان اجرای Visual Basic.NET است. از این زمان اجـرا با زبان Visual Basic.NET استفاده می شود. این فضانام همچنین دربرگیرنده کلاس هایی است که از کامپایل و تولید کد با استفاده از زبان ویژوال بیسیک پشتیبانی می کنند.

13-2. Microsoft.CSharp:

این فضای نام دربرگیرنده کلاس هایی است که از کامپایل و تولید کد بـا اسـتفاده از زبـان C# پشتیبانی می کنند.

13-3 . Microsoft.Jscript:

این فضای نام دربرگیرنده کلاس هایی است که از کامپایل و تولید کد بـا اسـتفاده از زبـان JScript پشتیبانی می کنند.

13-4. Microsoft.Win32:

کلاسها و رابطهای مورد نیاز برای کار با کلیدها و hiveهای رجیستری را تامین مـی کنـد. با وجود آنکه فضانام ها از قبل تامین می شوند، مـی توانیـد بـرای اسـتفاده از برنامـه کـاربردی ASP.NET فضانام های خود را ایجاد کنید. برای هر کلاس ایجاد شـده توسـط سـازنده یـک فضانام تولید می شود.

14- استفاده از فضانام ها در صفحات ASP.NET :

دو راه برای افزودن فضانام بـه برنامـه کـاربردی ASP.NET وجـود دارد. از شـبه دسـتور (Directive) صفحه Import@ برای صفحات ASPX و از کلمـه کلیـدی Imports بـرای افزودن فضانام به برنامه codebehind مربوطه در ویژوال بیسیک اسـتفاده مـی شـود و بـرای زبان C# از دستور using استفاده می گردد. قطعه برنامه زیـر ترکیـب نحـوی بـرای افـزودن فضانام System.Web.UI.WebControls به صفحه ASP.NET شما است.

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

همین فضانام را در قسمت codebehind بصورت زیر به برنامه اضافه می کنیم.

```
Imports System.Web.UI.WebControls (vb.net )
using System.Web.UI.WebControls; (C#)
```

به تفاوت Import و Imports دقت کنید

در صورتیکه می خواهید چند فضانام را به صفحه ASP.NET خود و یا صفحه codebehind اضافه کنید باید هرکدام را جداگانه اضافه کنید. بعنوان مثال، برای افزودن فضانام System.Web.UI.HTMLControls به صفحات با فضانام های 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 فضانام ها را ظاهر می کند، و می توانید به سادگی فضانام مورد نظر را با ماوس برگزینید.

امتیاز این فهرست آن است که مجبور نیستید همه فضانام های .NET را از حفظ بدانید، بلکه می توانید به آسانی از فهرست انتخاب کنید. این ویژگی با عنوان Intellisense شناخته می شود. برای صفحات ASP.NET از این ترکیب استفاده کنید:

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

برای صفحات codebehind ویژوال بیسیک از این ترکیب استفاده کنید:

```
Imports System.Web.UI.WebControls
Imports System.Web.UI.HTMLControls
Imports namespace
...
```

15- Smart Navigation چیست؟

مفهوم Smart Navigation و فواید آن :

Navigation Smart یکی از بهترین ابزارهای جدیدی است که ASP.NET آنرا عرضه کرده است. این ابزار جدید باعث شده ظاهر برنامه های وب و احساسی که نسبت به آن وجود دارد شباهت بیشتری با برنامه های عادی و نوشته شده برای ویندوز پیدا کند.

یکی از موانع بزرگ برنامه های تحت وب به معماری و ساختار HTTP برمی گردد. جاییکه مجبوریم اطلاعات جمع آوری شده در سمت مشتری را به سرور بازگردانیم.

به همین دلیل مجبور به رسم مجدد و کامل صفحه ای که قبلا دیده ایم می باشیم، که این نه تنها باعث می شود یک حالت فلش مانند در این رفت و برگشت و رسم مجدد رخ دهد، بلکه برای صفحه های بلند که برای دیدن تمام صفحه نیازمند بهscrolling هستیم، باعث می شود که دیدمان را به اول صفحه انتقال دهد، چیزی که هم شاید دلخواه ما نباشد و هم اینکه ممکن است باعث سردرگمی کاربر گردد. همچنین این فرآیند باعث تغییر فوکوس کنترل ها و بسیاری از اتفاقات دیگرنیز می شود.

در برنامه های عادی ویندوز ما به طور معمول فقط قسمت هایی از صفحه را به روز می کنیم که تغییری در آن ایجاد شده باشد یا تحت تاثیر چیزی قرار گیرند و این بدون نیاز به تغییر در کل برنامه می باشد (مثلا فقط یک عضو به listbox ما اضافه می شود. بدون تغییر و رسم مجدد فرم برنامه).

Smart Navigation یا به عبارتی هدایت هوشمندانه این توانایی موجود در برنامه های ویندوز را برای برنامه های تحت وب فراهم می کند! اما قبل از هر چیز باید بدانید که این ابزار فقط برای IE می باشد و آن هم نسخه های ۵ به بالاتر آن. با این وجود شما می توانید Smart Navigation را فعال یا غیرفعال سازید، بدون آنکه تاثیری در برنامه شما بگذارد.

حتی اگر شما در پروژه تان مرورگرهـای محتلفـی را مـدنظر قـرار داده باشـید، مـی توانیـد Smart Navigation را فعال سازید. در این صورت ASP.NET نـوع مرورگـر را تـشخیص داده و Smart Navigation را فقط برای مرورگرهای پشتیبانی شده فعال می سازد.

چهار مورد برجسته ای که Navigation Smart فراهم می کند عبارتند از:

• صفحه در میان درخواست ها یک نمایش ممتد را داراست و به عبـارتی حالـت فلـش زدن بـه خود نمی گیرد.

• موقعیت Scroll را حفظ می کند.

• فوکوس عضو دارنده فوکوس را نگه می دارد.

• آخرین صفحه درون تاریخچه (History) نگهداری می شود.

این ابزار در حالت واقع گرایانه برای برنامه هایی که ارسال به عقـب(!) Postback فراوانـی دارند طراحی شده است ولی با توجه به این نکته که محتوای صـفحه نبایـد زیـاد تغییـر نکنـد. احتمالا بنا به دلایل کارایی و نه اینکه در تغییرات زیاد ایرادی بهم بزند - مترجم.

شاید یک چیز شگفت آور در مورد این ابزار این باشد که شما در حقیقـت نیـاز بـه نوشـتن هیچ کد و برنامه ای ندارید.

نحوه استفاده:

Smart Navigation درون هدایت کننده صفحه (Page directive :)، برای تنظیم یک صفحه و درون web.config برای تنظیم کل برنامه استفاده می شود. برای تنظیم در Page Directive به صورت زیر عمل کنید:

<%@ Page SmartNavigation=true % >

و برای تنظیم در web.config از ساختار زیر استفاده نمایید:

< Configuration>
< System.web>

روش کار اینگونه است که کل صفحه بدرون یک فریم دورنی مخفی (hidden IFrame) بارگذاری (load) می شود و سپس فقط قسمت های تغییر کرده دوباره رندر(render) می‌شوند.

قسمت دیگر که می بایست توضیح داده شود بانک اطلاعاتی می باشد که از آن برای ذخیره اطلاعات استفاده می کنیم ، البته بحث زبان ASP.NET در اینجا به اتمام نمی رسد.

در اینجا ما به شرح قسمتی از این زبان پرداختیم ، در کل این زبان قسمتهای دیگری هم دارد که ما در اینجا به شرح و توضیح آن نمی پردازیم.

فصل دوم

1- معرفی SQL و دستورات عمومی آن

توسط SQL میتوان درون یک بانک اطلاعاتی پرس و جو کرده (Query) و نتیجه را برگرداند. بانک اطلاعاتی شامل آبجکتی به نام جدول (Table) میباشد. رکوردها در بانکهای اطلاعات در جداول ذخیره میگردند. جدول شامل سطر و ستون میباشند.

1-1.پر کاربرد ترین دستورات SQL شامل موارد زیر است

SELECT استخراج یک داده از بانک اطلاعاتی

UPDAT به روز رسانی یک داده درون بانک اطلاعاتی

DELETE پاک کردن یک داده از بانک اطلاعاتی

INSERT وارد کردن یک داده جدید به بانک اطلاعاتی

1-2. همچنین در SQL میتوان داده هایی نیز تعریف کرد

1-2-1. CREATE TABLE ایجاد یک جدول جدید

1-2-2. ALTER TABLE تغییر دادن یک جدول

1-2-3. DROP TABLE پاک کردن یک جدول

4-2-1. CREATE INDEX ایجاد یک اندیس

5-2-1. DROP INDEX (کلید جستجو) پاک کردن یک اندیس

2- SQL و Active Server Pages - ASP

SQL یکی از قسمتهای خیلی مهم ASP میباشد زیرا در ASP برای کار با بانکهای اطلاعاتی از SQL استفاده میشود. و توسط تکنولوژی ADO میتوان از SQL در ASP استفاده کرد.

1-1-1.دستور SELECT :

SELECT column-name(s) FROM table-name

مثال :

نام جدول Persons است

Last Name First Name Address City

Hasani Ali Esfahan

دستور Select همراه با شرط

SELECT column FROM table WHERE column condition value

عملگر ها در SQL

عملگر مفهوم

= معادل بودن

<> برابر نبودن

> بزرگتر

2-1-1.دستور DISTINCT (جداسازی)

SELECT DISTINCT column-name(s) FROM table-name

مثال :

نام جدول Order است Company Order Number

sega 3412

دستور Order By

برای مرتب کردن سطرها نام جدول Order است Company Order Number

sega 3412

ABS Shop 5678

w3s 3212

W3S 6778

مثال :

SELECT Company , OrderNumber

دستور Insert

INSERT INTO Table-Name (Column1 , Column2) VALUES

مثال :

DELETE FROM Person WHERE Lastname=’hasani’ نام جدول Persons

last name First name Adress city است

alian hasani NO 40 Esfahan

Hasani Ali No 15 Tehran

نتیجه :

last name First name Adress city

alian hasani NO 40 Esfahan

SEGA

W3S

Trio

3- دسترسی به یک پایگاه داده از یک صفحه ASP

- ایجاد یک ADO Connection به یک پایگاه داده
- باز کردن Connection پایگاه داده
- ایجاد یک Connection
- باز کردن ADO Record Set
- گرفتن داده هایی که نیاز داریم از Record set
- بستن Record Set
- بستن Record Set

3-1. ایجاد یک ADO Connection به یک پایگاه داده

روش DSN-LESS C:/InetPub/wwwroot/nor.mdb مسیر فایل نمونه < ٪ var

.conn=Server. CreateObject (‘‘ADODB

Connection”) Conn. Provider=” Microsoft . Jet . OLEDB.4.0” Conn.Open

(”C:/InetPub/wwwroot/nor.mdb“) ٪ >

3-2. روش ODBC

-۲- روش ODBC : در این روش ابتدا باید یک ODBD Connection به Data Base ایجاد کنیم و سپس از طریق ADO به فایل DSN به طریق زیر Connect کنیم.

var conn=Server. CreateObject ٪ >

ایجاد یک ODBC Connection به پایگاه داده MS Access

- وارد شدن به ODBC از Control Panel
- انتخاب System DSN
- کلیک کردن روی دکمه ADD
- انتخاب Microsoft Access Driver و کلیک کردن روی دکمه
- Finish در مرحله بعد کلیک کردن بر روی دکمه Select و تعیین محل پایگاه‌داده
- دادن یک نام در قسمت Data Source Name
- کلیک کردن روی دکمه OK برای اینکه قادر باشیم اطلاعات یک پایگاه داده را بخوانیم اطلاعات باید ابتدا در Record Set، Load شوند. بنابر این بعد از ساختن یک Connection باید یک Record Set ایجاد کنیم.

مثال :

Data Base = nor.mdb نام

نام جدول =

conn= ٪ > مسیر فایل Customer C:/Inetpub/wwwroot/nor.mdb Data Base

(”Server. CreateObject (“ADODB.Connection

Connection ایجاد

conn.Provider=”Micrisoft.Jet.OLEDB.4.0” conn . Open

rs=Server. CreateObject بازکردن (“C:/WebData/ nor.mdb”) Connection

(”(“ADODB.Recordset

Record Set ایجاد rs. Open(“Customer”, conn) Record Set بازکردن ٪ > در

این مثال تمام محتویات جدول Customer به rs ریخته میشود.

حال میخواهیم در انتخاب محتویاتی از Customer که میخواهند به rs انتقال داده شـوند از دستور SQL استفاده کنیم.

4,3-3. ساختن یک Connection و Record Set و به کار بردن SQL

"set conn=Server. CreateObject ("ADODB .Connection ٪ >

.conn. Provider="Microsoft.jet.OLEDB.4.0" conn

Open ("C:/Inetpub/wwwroot/nor.mdb") (Record Set)

5-3. گرفتن داده هایی که نیاز داریم از Record Set

بعد از اینکه Record Set را باز کردیم میتوانیم به داده هایی که نیاز داریم دسترسی داشته باشیم.

مثال :

دسترسی به فیلد name از جدول rs rs(name

چاپ داده ها :

مثال : چاپ فیلد name از جدول rs Response.write(rs(n

7,6-3. بستن Recordset و Connection :

- برای بستن Recordset rs.close()
- برای بستن Connection

4- خلاصه سازی داده ها

هنگامی که دو جدول در یک پیوند یک به چند سهیم هستند. وسیله محرک Query مقادیری از سطر را از یک طرف برای ارتباط دادن سطرها در طرفهای دیگری تکرار می کند. بعضی مواقع آن دقیقاً چیزی است که شما می خواهید، اما اغلب شما می خواهید تکرار کردن سطرها را از چندین طریق دسته بندی یا خلاصه سازید.

در این درس، ما به دو روش برای انجام آن که عبارتند از کلید واژه DISTINCT و شرط GROUP BY نگاه خواهیم کرد.

1-4. شناختن SELECT DISTINCT

یکی از اهداف طرح پایگاه داده ارتباطی برداشتن وابستگی داده ها می باشد. اما بیشتر پایگاه داده به طور ضروری مقادیر واقعی در چندین سطر را شامل خواهد شد. یک جدولی که شامل اطلاعات آدرس مشتری می باشد برای مثال احتمالاً کد ایالتی و کشوری برای چندین سطر خواهد داشت که نه اشتباه و نه تکراری می باشد.

نظر به اینکه هر کد ایالتی نسبتی از یک مشتری می باشد. بـه همـین نحـو یـک جـدول از چندین جهت از یک ارتباط یک به چند ممکن است هر مقدار کلید خارجی معلوم کـه چنـدین بار تکرار شده را داشته باشد. آن برای یک ارتباط یکپارچه از پایگاه داده لازم می باشد.

اگر چه این تکرار می توانـد بعـضی مواقـع نتـایج Query را نـامطلوب سـازد. یـک جـدول مشتری با ۱۰۰۰ سطر با ۹۰ درصد مشتری از کالیفرنیـا، Query زیـر کـد CA را ۹۰۰۰ بـار خواهد باز گرداند که اصلاً یک نتیجه مفیدی نمی باشد.

State From Customer SELECT

کلید واژه Distinct در این موقعیت شما را کمک می کند. Distinct کـه درسـت بعـد از SELECT قرار می گیرد به SQL Server دستور داده که سطرهایی چندگانه در قرارگیری نتایج را حذف نماید. بنابراین Query زیر هر کد ایالتی را فقط یک بار باز می گرداند بـه طـور وضوح لیستی که شما جستجو می کنید.

State From Customer SELECT DISTINCT

راهنمـایی: همتـای کلیـد واژه Distinct، All مـی باشـد کـه Server SQL را بـرای بازگرداندن همه سطرها آگاه می سازد خواه آن واحـد باشـد یـا خیـر. از مـوقعی کـه ایـن یـک عملکرد پیش فرض از یک عبارت SELECT می باشد All بـه طـور معمـول اسـتفاده نمـی گردد. اما شما ممکن تصمیم بگیرید به در برداشتن آن اگر سـاختار دسـتور Query را بیـشتر قابل فهم سازید.

2-4. استفاده کردن از SELECT DISTINC:

کلیـد واژه Distinct مـی توانـد در عبـارت SQL از Query Distinct یـا بـه وسـیله تنظیمات Properties از Query مشخص گردد.

1-2-4. ایجاد کردن Distinct Query با استفاده از قاب دیاگرام

۱ -Query Designer را برای جدول Oils به وسیله کلیک راست کردن نام جدول در قاب Details باز کرده روی جدول Open رفته و همه سطرهای بازگشتی را انتخاب می کنیم.

۲- قاب دیاگرام را به وسـیله کلیـک کـردن روی دکمـه قـاب دیـاگرام در نـوار ابـزار Query Designer نشان می دهیم.

۳- دکمـه Add Table را کلیـک مـی کنـیم. Designer Query کـادر محـاوره ای AddTable را نشان می دهد.

۴- Plantparts را در لیست جدول انتخاب مـی کنـیم و سـپس Add را کلیـک مـی کنـیم. Query Designer جدول را به Query اضافه می کند.

۵- Close را برای بستن کادر محاوره ای کلیک می کنیم.

۶- دکمه قاب SQL را در نـوار ابـزار Query Designer کلیـک مـی کنـیم. قـاب SQL، Query Designer را نشان می دهد.

۷- علامت × بعد از کلید واژه SELECT را حذف می کنیم.

۸- دکمـه قـاب SQL را در نـوار ابـزار Query Designer کلیـک مـی کنـیم (OK را اگـر Query Designer یک متن خطا درباره ساختار دستور عبارت SELECT نشان مـی دهـد کلیک می کنیم). Query Designer قاب SQL را پنهان می سازد.

مهم: زمانی که شما Designer Query را باز می کنید حالت SQL پیش فـرض معمـولاً SELECT× می باشد. ستون ویژه در قاب دیاگرام به سبب اینکه آنها به لیست سـتون اضـافه می شوند انتخاب می گردند. مایکروسافت این را یک ویژگی در نظر می گیرد.

۹- در قاب دیاگرام فقط ستون Plantpart را از جدول Plantparts برای نشان دادن انتخـاب می کنیم.

۱۰- دکمه Run را برای اجرای مجـدد Query کلیـک مـی کنـیم. Query Designer هـر مقدار Plantpart را چندین بار لیست می کند.

۱۱- در یک ناحیه خالی از قاب دیاگرام کلیک راست کرده و Properties را انتخاب می کنیم. Query Designer کادر محاوره ای Properties را نشان می دهد.

۱۲- گزینه مقادیر Distinct را انتخاب می کنیم.

۱۳- Close را برای بستن کادر محاوره ای کلیک می کنیم.

۱۴- دکمه Run را برای اجرای مجـدد Query کلیـک مـی کنـیم. Query Designer هـر مقدار را فقط یک بار نشان می دهد.

2-2-4. ایجاد کردن یک SELECT DISTINCT با استفاده از قاب SQL

۱- قاب دیاگرام را پنهان ساخته و قاب SQL را نشان می دهیم.

۲- عبارت SELECT موجود را با متن زیر جایگزین می کنیم.

```
SELECT DISTINCT Plant Types Plant Type
INNER Join FROM Oils
Plant Types ON Oils Plant Type ID=Plant Types Plant Type ID
```

۳- دکمه Run را برای اجرا مجـدد Query کلیـک مـی کنـیم. Query Designer مقـدار PlantType متمایز که به وسیله جدول Oils بازگشت شده را نشان می دهد.

3-4. شناختن GROUP BY

کلید واژه Distinct امر می کند. SQL Server را به بازگرداندن سطرهای واحد نظر به اینکه شرط GROUP BY، SQL Server را به ترکیب سطرها با مقادیر یکسان در ستون یا ستونهای مشخص شده در شرط در داخل سطر تکی امر می کند.

مهم: هر ستونی که در شرط GROUP BY شامل می باشد باید در خروجی Query شامل گردد.

شرط GROUP BY اغلب با یک aggregate Function استفاده می گردد. یک Function aggregate محاسبات در مجموعه از مقادیر را به انجام می رساند و یک نتیجه تک مقداری را باز می گرداند. رایج ترین گزینه های به هم پیوسته استفاده شده در پرس و جوهای GROUP BY، MIN می باشد که کوچکترین مقادیر را در مجموعه باز می گرداند. MAX که بزرگترین مقادیر در مجموعه را باز می گرداند و COUNT که تعدادی از مقادیر را در یک مجموعه باز می گرداند.

استفاده کردن از GROUP BY

شرط GROUP BY می تواند مشخص شود با استفاده کردن از هر یک از قابها در Query Designer ، اما قابهای SQL و Grid بیشترین کنترل را فراهم می سازد.

1-3-4. ایجاد کردن یک GROUP BY Query با استفاده از قاب Grid

۱- قاب SQL را پنهان ساخته و قاب Grid را نشان می دهیم.

۲- ستون OilName را برای Query اضافه می کنیم.

۳- دکمه Group By را در نوار ابزار Query Designer کلیک می کنیم. Query Designer یک ستون Group By برای شبکه اضافه می کند و هر دوی مقادیر را برای Group By قرار می دهد.

۴- سل Group By را برای تغییر دادن سطر OilName به Count تغییر می دهیم.

۵- دکمه Run را برای اجرا مجدد Query کلیک می کنیم. Query Designer تعدادی از Oils را برای هر PlantType نشان می دهد.

2-3-4. ایجاد کردن یک Group By Query با استفاده از قـاب SQL :

۱- قاب شبکه را پنهان ساخته و قاب SQL را نشان می دهیم.

۲- عبارت SELECT موجود را با متن زیر جایگزین می کنیم.

part Count (Oils Oil Name) As Number Of SELECT Plant parts Plant Oils
FROM Oils INNER Join
parts ON Oils Plant part ID=Plant parts Plant part ID Plant
Plant part GROUP BY Plant parts

۳- دکمه Run را برای اجرای مجدد Query کلیک مـی کنـیم. Designer Query مقـدار oils برای هر Plantpart نشان می دهد.

4-4. استفاده کردن از شرط HAVING

شرط Having سطرهای بازگردانده شـده بـه وسـیله شـرط GROUP BY را از همـان راهی که یک شرط Where سطرهای بازگردانده شده را به وسیله شـرط SELECT محـدود می سازد.

هر دو شرط Where و Having می تواند در یـک عبـارت SELECT شـامل شـود کـه شرط Where قبل از گروه بندی علمیات و شرط Having بعد از آن به کار بـرده مـی شـود.

ساختار دستور شرط Having همانند شـرط Where مـی باشـد بـه اسـتثنای اینکـه شـرط Having می تواند شامل یکی از توابعهای به هم پیوسته باشد که در لیـست سـتونها از شـرط SELECT قرار می گیرد.

اگر چه شما باید توابع به هم پیوسته را تکرار کنید. برای مثال شرط Having که در حالت زیر استفاده شده صحیح می باشد.

```
Name) As Number Of  SELECT Plant parts Plant part Count (Oils Oil
Oils
FROM Oils INNER Join
part ID=Plant Parts Plant part ID Plant parts ON Oils Plant
GROUP BY Plant parts Plant part
<(Count (Oils Oil Name HAVING
```

اگر چه شما نمی توانید از اسم مستعار برای تابع Count در شرط Having استفاده کنید. بنابراین شرط Having زیر درست نخواهد بود.

```
<HAVING Number Of Oils
```

1-4-4. ایجاد کردن یک Query با استفاده از HAVING در قاب Grid

۱- قاب SQL را پنهان ساخته و قاب Grid را نمایش می دهیم.

۲- Add۵> را برای سل مورد نظر از ستون oil Name قرار می دهیم.

۳- دکمه Run در نوار ابزار Query Designer را برای اجـرای مجـدد Query کلیـک مـی کنیم.

2-4-4. ایجاد کردن یک Query با استفاده از HAVING در قاب SQL:

۱- قاب Grid را پنهان ساخته و قاب SQL را نشان می دهیم.

۲- شـرط Having را بـرای HAVING (Count (Oils Oil Name)>۵) تغییـر مـی دهیم.

۳- دکمه Run در نوار ابزار Query Designer را برای اجـرای مجـدد Query کلیـک مـی کنیم. Query Designer فقط آن Plantparts که Oils اشتراکی کمتر از ۵ دارد را نـشان می دهد.

5- مرتب سازی و انتخاب کردن سطرها

در فصل گذشته ما بیشترین فرمهای مقدماتی از حالت SELECT را مرور کـرده و آنهـا را برای انتخاب ستونهای یک جدول استفاده کردیم. اما بیشتر مواقع شما بازگشت بـه سـطرها در جدول پس زمینه یا نمایش در یک قاعده خاص و فقط بازگشت به یک زیر مجموعـه از آنهـا را خواهید خواست. شرطهای ORDER BY و WHERE در این درس بررسی شده و اجـرای آن برای شما فراهم شود.

الف- شرط ORDER BY

شرط The ORDER BY یک ترکیب گزینه ای از یک حالت SELECT می باشـد. آن به شما برای مشخص کردن ترتیب در آن سطرهایی که بازگشت خواهند کرد اجازه مـی دهـد. ستونهای چندگانه می توانند مشخص شوند و سطرها می توانند به صورت ترتیـب افزایـشی یـا کاهشی باز گردانده شوند.

5-1. مرتب سازی سطرها

ساده ترین فرم در شرط ORDER BY فراهم می سازد یک نـام سـتون تکـی کـه بـرای مرتب کردن سطرها که به وسیله Query باز گردانده می شود استفاده خواهد شد.

1-1-5. مرتب کردن سطرها با استفاده از قاب Grid:

- Query Designer را برای جدول Oils به وسیله کلیک راست کردن نام آن در قاب Details باز می کنیم. به زیر منوی جدول Open رفته و همه ستونهای بازگشتی را انتخاب می کنیم. SQL Server، Query Designer را برای جدول باز می کند.

- قاب Grid را به وسیله کلیک راست کردن دکمه قاب Grid در نوار ابزار Query Designer نشان می دهیم.

- فقط ستونهای OilID، Oil Name، LatinName را برای نمایش انتخاب می‌کنیم. Query Designer محتویات قاب Results را که کمرنگ شده برای نشان دادن اینکه با مشخصات Query زیاد مربوط نمی باشد نشان می دهد.

- دکمه Run را در نوار ابزار Query Designer برای اجرای Query کلیک می‌کنیم. Query Designer فقط ستونهای مشخص شده را نشان می دهد.

- نوع Sort فیلد Oil Name را صعودی قرار می دهیم.

- دکمه Run را در نوار ابزار Query Designer برای اجرای Query کلیک می‌کنیم. Query Designer سطرهای مرتب شده با Oil Name را نشان می دهد.

2-1-5. مرتب سازی سطرها با استفاده از قاب SQL

- قاب Grid را پنهان ساخته و قاب SQL را بـه وسـیله کلیـک کـردن دکمـه هـا روی نوارابزار Query Designer نشان می دهیم.

- DESC را بعد از شرط ORDER BY OilName اضافه می کنیم.

راهنمایی: کلید واژه DESCبه SQL Server برگرداندن سطرها به ترتیب نزولـی را ابـراز می کند. کلید واژه ASC که گزینه ای می باشد سطرها را به ترتیب صعودی برمی گرداند.

- دکمـه Run را روی نـوار ابـزار Query Designer بـرای اجـرای Query کلیـک می‌کنیم. Query Designer نتایج را کـه بـا OilName ذخیـره شـده بـه صـورت نزولی نشان می دهد.

2-5. مرتب سازی ستونهای چندگانه

شما می توانید ستونهای چندگانه را در شرط ORDER BY مـشخص کنیـد. زمـانی کـه ستونهای چندگانه مشخص می گردند ترتیب سـتونها نتـایج SQL Server را کـه بـه وسـیله اولین ستون و سپس به وسیله دومین ستون و بنـابراین چهـارمین سـتون مرتـب خواهـد شـد مشخص می کند.

راهنمایی: تمرینها در این بخش از جدول OilOdors استفاده می کند که به صورت جدول الحاقی انجام وظیفه می کند که ارتباط چندبه چند بین جدول Oils و Odors را تجزیه و تحلیل می کند. معمولاً شما کلیدهای خارجی ترکیب شده در این جدول را با استفاده از یک ارتباط تجزیه می کردید.

1-2-5. مرتب سازی ستون ها با استفاده از قاب Grid

- پنجره شماره ۱ را از منوی Window برای برگشت به درخت Console انتخاب می‌کنیم.

- Query Designer را برای جدول Oil Orders به وسیله کلیک راست کردن نام آن در قاب Details باز کرده روی زیر منوی جدول Open رفته و همه سطرهای بازگشتی را انتخاب می کنیم. SQL Server، Query Designer را برای جدول باز می کند.

- قاب Grid را به وسیله کلیک کردن دکمه قاب Grid روی نوار ابزار Query Designer نشان می دهیم.

- × را در قاب Grid با نامهای دو فیلد جایگزین می کنیم. SQL Server محتویات از قاب Results را برای نشان دادن اینکه آن با مشخصات Query زیاد در ارتباط نیست کم رنگ می کند.

- دکمه Run را در نوار ابزار Query Designer برای اجرای Query کلیک می‌کنیم. Query Designer فقط ستونهای مشخص شده شما را نشان می دهد.

- نوع Sort هر دوی ستونها را صعودی قرار می دهیم.

- دکمه Run را در نوار ابزار Query Designer برای اجرای Query کلیک می‌کنیم. Query Designer سطرهای مرتب شده را با اولین OilID و سپس با OdorID در داخل OilID نشان می دهد.

5-2-2. مرتب کردن ستون ها با استفاده از قاب SQL

- قاب Grid را پنهان کرده و قاب SQL را به وسیله کلیک کردن دکمه ها در نوار ابزار Query Designer نشان می دهیم.

- ستونها را در شرط ORDER BY معکوس می کنیم.

- دکمه Run را در نوار ابزار Query Designer برای اجرای Query کلیک می‌کنیم. Query Designer نتایج مرتب شده را با اولین OdorID و سپس با OilID نشان می دهد.

- پنجره Query Designer را می بندیم.

ب- شرط WHERE

با استفاده از شرط اختیاری WHERE از حالت SELECT شما می توانید یک زیر مجموعه از سطرها که باز گردانده می شوند را مشخص کنید. برای مثال شما ممکن است بخواهید فقط مشتریهایی که بیش از ۱۰۰۰$ در ۱۲ ماه قبل خرج کرده اند را ببینید یا اینکه فقط نامهای Oil که با حرف R شروع می شوند را ببینید. شما این ملاکها را با استفاده از شرط WHERE مشخص خواهید کرد.

3-5. شرط WHERE BASIC

Operator Meaning

مساویست با = بزرگتر از

< کوچکتر از >بزرگتر یا مساویست با =<کوچکتر یا مساویست با => مساوی نیست با <>

کلید برای شرط WHERE یک ملاک انتخابی می باشد که مشخص می کند که کدام سطرها باز خواهند گشت. ساختار پایه ای از یک شرط WHERE، WHERE می باشد.

مشخص شده در شرایط WHERE می توانـد یـک ارزش دائمـی باشـد ماننـد Red″ یـا ۱۰۰۰۰ یا می تواند باشد یک عبارتی که یک ارزش مانند GETDATE. را باز گرداند.

به طور شبیه ارزش می تواند دستی ساخته شوند با استفاده از تابعهـای Transact-SQL مانند LEFT که یک تعدادی از کاراکترهای مشخص شده از چپ یک رشته را باز می گرداند.

1-3-5. مشخص کردن یک شرط WHERE با استفاده از قاب Grid :

- پنجره شماره ۲ را از منوی Window برای بازگشت به پنجـره Query Designer که ما زودتر در این درس استفاده کرده ایم انتخاب می کنیم.

- قاب SQL را پنهان کرده و قاب Grid را به وسیله کلیک کردن دکمه ها در نوار ابزار Query Designer نشان می دهیم.

- ″Eucalyptus=″ را در سل ملاک سطر OilName از قاب Grid وارد می کنیم.

- دکمـه Run را روی نـوار ابـزار Query Designer بـرای اجـرای Query کلیـک می‌کنیم، Query Designer فقط یک سطر تنها را نشان می دهد.

2-3-5. مشخص کردن یک شرط WHERE با استفاده از قاب SQL:

- قاب Grid را پنهان ساخته و قاب SQL را به وسیله کلیک کردن دکمه ها در نوارابزار Query Designer کلیک می کنیم.

- شرط WHERE را با WHERE (LEFT(Oil Name,1)="R″) تغییر می‌دهیم.

- دکمه Run را در نوار ابزار Query Designer برای اجرای Query کلیک می

- کنیم. Query Designer نامهای Oil را که با ″R″ شروع می گردد را نشان می‌دهد.

4-5. استفاده کردن از اپراتورهای ویژه

علاوه بر فرمت استاندارد برای یک شرط WHERE از ، SQL Server همچنین سه عملگر را پشتیبانی می کند: LIKE که اجازه می دهد به شما به فراهم ساختن مقدار نامشخص با استفاده از وایلد کارتهای نشان داده شده در جدول ۲-۱۳ و BETWEEN که اجازه می دهد به شما مشخص کردن یک محدوده از ارزشها و IN که اجازه می دهد به شما برای مشخص کردن یک مجموعه از مقادیر.

مثال معنی Wildcard

'LIKE 'a که ارتباط می دهد ″at″ و ″as″ هر تک کـاراکتر -امـا نـه ″ILKE%'t'%"and که ارتباط می دهد ″at″ و ″bat″ و ″Together″ اما نه ″Lucky″ هر رشته از صـفر یـا چنـدین کاراکتر ٪'LIKE '[a-c]at که ارتباط می دهد″Cat″ و ″bat″ اما نه ″fat″ هر کـاراکتر ویـژه در داخـل یـک محـدوده یـا مجموعـه []'LIKE '[a-b]at کـه ارتبـاط مـی دهـد″Bat″ امـا نـه ″[^c]at' که ارتباط می دهد ″Bat″ و ″fat″ اما نه ″Cat″ هر کاراکتر ویژه کـه در داخل یک محدوده یا مجموعه نمی باشد .

1-4-5. مشخص کردن یک شرط WHERE با استفاده از LIKE

- شـرط WHERE را در قـاب SQL بـرای WHERE تغییـر مـی دهـیم.)٪(Oil 'Name LIKE 'Rose

- دکمه Run را در نوار ابزار Query Designer برای اجرای Query کلیک می‌کنیم. Query Designer همه سطرهایی که با Rose شروع می گردد را نشان می‌دهد.

2-4-5. مشخص کردن یک شرط WHERE با استفاده از BETWEEN:

- شـــــرط WHERE را در قـــــاب SQL بـــــا WHERE(LEFT(Oil Name,1)BETWEEN A AND 'C') تغییر می دهیم.

- دکمه Run را در نوار ابزار Query Designer برای اجرای Query کلیک می‌کنیم. Query Designer همه سطرها که با A و B یا C شروع می گردد را نشان‌می‌دهد.

راهنمایی: Transact-SQL همچنـین عملگرهـا Not BETWEEN را پـشتیبانی مـی کند که دقیقاً از همان طریق کار می کند. بـه اسـتثنای اینکـه آن شـامل یـک محـدوده ای از مقادیر می باشـد. بـرای مثـال 'Left (Oil Name , 1) Not BETWEEN 'C' And 'E همه سطرها باز خواهد گرداند به جز آنهایی که Oil Name با C و D یا E شروع می کند.

3-4-5. مشخص کردن یک شرط WHERE با استفاده از IN

- شـــرط WHERE را در قـــاب SQL بـــا WHERE (LEFT (OilName , 1) ('IN('G','M','V تغییر می دهیم.

- دکمه Run را در نوار ابزار Query Designer برای اجرا Query کلیک می کنیم. Query Designer، 8 سطر را نشان می دهد.

5-5. ترکیب کردن ملاک انتخابی

علاوه بر مشخص کردن یک شرط WHERE با استفاده از عبارت Format شما می‌توانید همچنین عبارات را با استفاده از عملگر منطقی OR یا AND ترکیب کنید.

یک شرط WHERE از فرمت FormatOR سطرهایی که به ملاک مربوط می گردند خواهد بازگرداند، نظر به اینکه یک شرط WHERE از And Format سطرهایی که به هر دو ملاکها مربوط می گردند را باز خواهد گرداند.

5-5-1. مشخص کردن ملاک پیچیده با استفاده از OR:

- شرط WHERE رادرقاب SQL به WHERE (OilName='Rosemary' OR OilName='Sagy) تغییر می دهیم.

- دکمه Run را روی نوار ابزار Query Designer برای اجرای Query کلیک می‌کنیم. Query Designer دو سطر را نشان می دهد.

2-5-5. مشخص کردن ملاک پیچیده با استفاده از AND

- شرط WHERE را در قاب SQL به (Oil Name LIKE 'Rose' WHERE٪) تغییر می دهیم.

- دکمه Run را روی نوار ابزار Query Designer برای اجرای Query کلیک می‌کنیم. Query Designer، 4 سطر را نشان می دهد.

- شرط WHERE را در قاب SQL به (OilName%) And (OilID<40) LIKE Rose WHERE تغییر می دهیم.

- دکمه Run را روی نوار ابزار Query Designer برای اجرای Query کلیک می‌کنیم. Query Designer سطر Roso Otto را شامل می گردد.

در فصل بعدی به شرح زبان VB.NET زبانی که در کل پروژه مورد استفاده قرار گرفته است می پردازیم.

فصل سوم

1- تاریخچه ویژوال بیسیک

قبل از معرفی ویژوال بیسیک، پیاده کنندگان نرم افزار مجبور به تسلط و مهارت در زمینه استفاده از C++ به همراه موارد پیچیده ای در این خصوص بودند. بدین ترتیب، صرفاً افراد خاص آموزش دیده، قادر به خلق نرم افزارهای قدرتمند به منظور اجرا در محیط ویندوز بودند.

ویژوال بیسیک، محدودیت فوق را تغییر و می توان این ادعا را داشت که امروزه خطوط زیادی از برنامه های نوشته شده با استفاده از ویژوال بیسیک کد شده است.

ویژوال بیسیک، ظاهر برنامه نویسی تحت ویندوز را با حذف عملیات اضافی برای نوشتن کدهای لازم جهت طراحی بخش رابط کاربر (UI)، تغییر داده است.

در این راستا زمانی که بخش رابط کاربر، ترسیم می شود، برنامه نویس می تواند کدهای لازم به منظور انجام عکس العمل مناسب در رابطه با رویداد ها را به آن اضافه نماید. زمانی که ماکروسافت نسخه شماره سه ویژوال بیسیک را ارائه کرد، مجدداً دنیای برنامه نویسی با تغییر مهمی مواجه شد. در این راستا امکانات مناسبی برای نوشتن برنامه های مبتنی بر بانک های اطلاعاتی، در اختیار برنامه نویسان قرار گرفت.

مایکروسافت بدین منظور محصول جدیدی با نام (Datat Access Objects DAO) را ارائه نمود. برنامه نویسان با استفاده از DAO امکان انجام عملیات متفاوت در رابطه با داده ها را به دست آوردند. نسخه های شماره چهار و پنج، قابلیت های نسخه سه را افزایش و این امکان را برای پیاده کنندگان نرم افزار فراهم نمود تا برنامه های خود را جهت اجرا در محیط ویندوز ۹۵ طراحی و پیاده سازی نمایند.

در این زمینه برنامه نویسان قادر به نوشتن کدهایی گردیدند که امکان استفاده از آنان توسط سایر پیاده کنندگان نرم افزار که از زبانی دیگر استفاده می کردند، فراهم شد.

نسخه شماره شش ویژوال بیسیک، روش جدیدی به منظور دستیابی به بانک های اطلاعاتی را ارائه کرد: ActiveX)ADO Data Objects) .

یکی از اهداف اولیه طراحی ADO امکان دستیابی بـه بانـک هـای اطلاعـاتی بـرای پیـاده کنندگان برنامه های مبتنی بر وب است که از تکنولوژی ASP استفاده می نمایند.

همزمان با ارائه جدیدترین نسخه ویژوال بیسیک که VB.NET نامیده می شـود، بـسیاری از محدودیت های مرتبط با ویـژوال بیـسیک برطـرف گردیـد. در گذشـته ویـژوال بیـسیک بـا انتقادات فراوان مواجه شد (عدم وجود امکانات مناسب در مقایسه با جاوا و یا C++) و بسیاری آن را نظیر یک اسباب بازی در دنیای وسیع زبان های برنامه نویسی می پنداشتند.

VB.NET با غلبه بر مشکلات نسخه های پیشین، توانسته است در مدت زمان کوتاهی، به عنوان یک ابزار پیاده سازی بسیار قدرتمند مطرح و گزینه ای مناسب بـرای برنامـه نویـسان در تمامی سطوح باشد.

2- مقدمه‌ای بر برنامه نویسی شیءگرا

برنامه نویسی در محیط NET. بر پایه اشیاء انجام می‌گیرد. اشیا طرحهایی برنامه‌پذیرند که می‌توانند نمونهایی از بسته‌های از دادهای مرتبط و دستورات باشـند. اشـیاء طرحهـای کامـل و خاصی برای دیگر اعضای محیط برنامه هستند، بدون این که جزییات کارهای درونی خود شیئ مطرح شود.

اشیاء از یک قالب به نام کلاس ساخته می‌شوند. کلاسهای کتابخانه‌ای پایـه‌ای NET. یـک سری کلاسهایی برای درست کردن اشیاء در برنامه‌هایتان آماده کرده‌اند.

همچنین شما می‌توانید کلاسهای اختصاصی خودتان را نیز ساخته و استفاده کنید. در این مقاله شما با مقدمات برنامه نویسی شیئ گرا آشنا می‌شوید.

اشیاء، اعضا و تجرد (Abstraction)

یک شیئ (Object) یک طرح برنامه‌پذیر است که چیزهایی را نشان می‌دهد.

در دنیای واقعی، ماشین، دوچرخه، کامپیوتر و... شیئ هستند.

هر کدام از این اشیاء یک سری اعمال و خصوصیات دارند. در برنامه شما، یک شیئ شاید یک فرم یا یک کنترل مانند یک دکمه (Button) یا یک Database Connectin یا هر چیز دیگری باشد.

هر شیئ یک واحد عملیاتی کامل است، و شامل همه داده‌های مورد نیازش و دارای همه اعمالی که برای آن ساخته شده است می‌باشد.

3- کلاس‌ها (Classes)

کلاسها قالب‌هایی برای اشیاء هستند. کلاسها را می‌توان، "طرحهای اولیه" برای اشیاء فرض کرد. آنها تمام عضوهای یک شیئ را تعیین، و رفتارهای آن را نیز تعریف می‌کنند.

وقتی که یک کلاس مقداردهی شد، یک نمونه درون حافظه‌ای از آن کلاس ساخته می‌شود. این نمونه شیئ نامیده می‌شود. برای نمونه سازی از کلاس از کلمه کلیدی New استفاده می‌شود.

```
Declares a variable of the Widget type '
Dim myWidget As Widge
Instantiates a new Widget object and assigns it to the myWidget '
Variable
()myWidget = New Widget
```

وقتی یک نمونه از یک کلاس ساخته می‌شود، یک کپی از نمونه داده بوسیله آن کلاسی که در حافظه ساخته شده تعریف می‌شود و به مرجع متغییر داده می‌شود.

هر نمونه از کلاس مستقل از دیگر نمونه هاست و می‌تواند یک طرح جداگانه برنامه پذیر باشد. درهر لحظه، محدودیتی برای تعداد کپی های یک کلاس (که قبلا تعریف شده) وجود ندارد.

برای مقایسه، در جهان واقعی، اگر ماشین یک شیئ باشد، کلاس یک طرح برای ماشین است. یک طرح می‌تواند برای هر تعداد ماشین مورد استفاده قرار گیرد و تغییرات بر روی یک ماشین، تاثیری بر دیگر ماشین‌ها نمی‌گذارد.

4- اشیاء و اعضاء (Members)

اشیاء ترکیبی از عضوها هستند. اعضا، تشکیل شده از خصوصیات (Properties)، فیلدها، متدها و رویدادها (Events) و هر چیزی که اطلاعات و اعمالی داشته باشد. فیلدها و خصوصیات، اعضای داده‌ای هر شیئ هستند. متدها اعمالی هستند که شیئ می‌تواند انجام دهد و رویدادها اطلاعاتی هستند که زمانیکه اتفاقی در برنامه می‌افتد یک شیئ به اشیاء دیگر می‌فرستد یا از آنها دریافت می‌کند.

در مثال واقعی‌مان، شیئ ماشین، فیلدها و خصوصیاتی مانند "رنگ"، "مدل"، "سال تولید" دارد. این اطلاعات وضعیت شیئ ماشین را توصیف می‌کنند. شیئ ماشین می‌تواند متدهایی مانند "دور زدن" و "تعویض دنده" داشته باشد.

متدها رفتاری که شیئ می‌تواند اجرا کند را نشان می‌دهند. ماشین شاید رویداد EngineOverheating از طرف شیئ "موتورش" را داشته باشد، یا وقتی با شیئ "درخت" تعامل می‌کند، شاید رویداد "تصادف" برای آن اتفاق بیفتد.

5- کپسوله سازی (Encapsulation)

کپسوله سازی روشی است که یک شیئ را مستقل از اینترفیسش پیاده سازی کنیم. یک برنامه با یک شیئ بواسطه اینترفیسش تعامل می‌کند، که شامل خصوصیات عمومی و متدهایش است.

تا زمانی که این اینترفیس ثابت باقی می‌ماند، برنامه می‌تواند به تعامل با کامپوننت ادامه دهد؛ حتی اگر پیاده سازی اینترفیس بین دو نسخه کاملا" بازنویسی شده باشد.

اشیاء فقط از طریق متدها و خصوصیات عمومیشان با دیگر شیئ‌ها تعامل می‌کنند. داده‌های داخلی یک شیئ، نباید در اینترفیس قرار بگیرد. بنابراین فیلدها به‌ندرت Public تعریف می‌شوند.

به مثال ماشینمان برگردیم:

اگر شیئ ماشین با شیئ راننده تعامل کند، اینترفیس ماشین شاید شامل متدهای "حرکت به جلو"، "حرکت به عقب" و "توقف" باشد. این همه اطلاعاتی است که راننده برای تعامل با ماشین نیاز دارد. ماشین شاید شامل شیئ "موتور" نیز باشد، اما راننده نیازی به شناخت شیئ موتور ندارد.

همه اطلاعاتی که راننده درباره این متدها دارد این است که می‌توانند فراخوانی شوند و مقادیر ویژه‌ای نیز را برمی‌گردانند. بنابراین اگر شیئ موتور تغییری کند، تا زمانیکه اینترفیس به درستی به کار خود ادامه می‌دهد این امر تفاوتی برای راننده ایجاد نمی کند.

6- چند شکلی (Polymorphism)

چند شکلی توانایی کلاسهای متفاوت، در پیاده سازی‌های مختلف از اینترفیس‌های عمومی مشابه است. به عبارت دیگر، چندشکلی به متدها و خصوصیات یک شیئ اجازه می‌دهد، بدون توجه به چگونگی پیاده سازی اعضای آنها، فراخوانی شوند. برای مثال شیئ Driver می‌تواند بوسیله اینترفیس عمومی ICar با شیئ Car تعامل کند. اگر شیئ دیگری مانند شیئ Truck یا شیئ SportCar اینترفیس عمومی مشابهی را داشته باشند، شیئ Driver می‌تواند با آنها

بدون توجه به پیاده سازی خاص آن اینترفیس تعامل کند. در اینجا دو راه اصلی برای تامین چندشکلی وجود دارد:

چندشکلی اینترفیسی (Interface Polymorphism)

چندشکلی وراثتی (Polymorphism Inheritance)

چندشکلی اینترفیسی (Interface Polymorphism)

اینترفیس یک قرارداد برای رفتار است. در واقع اینترفیس، اعضای یک کلاس را تعیین می‌کند، اما توضیحاتی درباره پیاده سازی جزییات آن نمی‌دهد. یک شیئ می‌تواند اینترفیس‌های زیاد و متفاوتی را پیاده سازی کند و کلاسهای متفاوت زیادی می‌توانند یک اینترفیس مشابه را پیاده سازی کنند. همه اشیایی که اینترفیس مشابهی را پیاده سازی می‌کنند می‌توانند با دیگر اشیاء درون اینترفیس تعامل کنند.

به عنوان مثال شیئ Car مثال قبل شاید اینترفیس IDrivable را پیاده سازی کند (به عنوان قرار داد، اینترفیس‌ها با I شروع می‌شوند)، که متدهای GoForward, GoBackward و Stop را تعیین می‌کند. کلاس‌های دیگر مانند Forklift،Truck یا Boat می‌توانند این اینترفیس را پیاده سازی کنند و بنابراین می‌توانند با شیئ Driver تعامل داشته باشند. شیئ Driver از پیاده سازی انیترفیسی که با آن تعامل دارد بی اطلاع است.

1-6. چندشکلی وراثتی (Inheritance Polymorphism)

وراثت به شما امکان می‌دهد که اعمال یک کلاس از پیش تعریف شده را در یک کلاس جدید با هم ترکیب کنید و اعضای متفاوت مورد نیاز را در آن پیاده سازی کنید. کلاسی که از کلاس دیگری ارث می برد "مشتق" آن کلاس، یا "وارث" آن کلاس نامیده می‌شود.

یک کلاس می‌تواند مستقیما" فقط از یک کلاس ارث ببرد، که آن کلاس که از آن ارث می برد را کلاس پایه (Class Base) می‌نامند. کلاس جدید اعضای مشابهی با کلاس پایه دارد، و اعضای اضافی که مورد نیاز باشند می‌توانند افزوده شوند. به علاوه، در کلاس جدید بوسیله overriding پیاده سازی کلاس پایه، اعضای پایه می‌توانند تغییر کنند.

کلاسهای وارث، همه ویژگی‌های کلاس پایه را نگه می‌دارند و می‌توانند با دیگر اشیاءیی که نمونه‌هایی از کلاس پایه هستند تعامل کنند.

به عنوان مثال، اگر کلاس Car یک کلاس پایه باشد، SportsCar می‌تواند یک کلاس مشتق شده از آن باشد. کلاس SportsCar نیز می‌تواند کلاس پایه‌ای برای کلاس مشتق شده دیگری مثل کلاس ConvertibleSportsCar باشد. هر کلاس مشتق شده جدید، ممکن است اعضای جدید را پیاده سازی کند، اما اعمالی که در کلاس اولیه Car تعریف شده‌است همچنان باقی می‌مانند.

7- Overloading اعضاء

Overloading به شما امکان می‌دهد که چندین عضو با نام مشابه درست کنید. هرعضوی که همنام با عضو دیگری است، باید امضای متفاوتی داشته باشد. Overloading بیشتر در بین متدها متداول است.

شاید شما بخواهید متدی درست کنید که بتواند مجموعه‌های متفاوتی از پارامترها را بپذیرد:

```
(Public Sub Display(ByVal DisplayValue As Integer
Omitted Implementation '
End Sub
```

این متد کاملا" پذیرفتنی است. اما فرض کنیم که می‌خواهید به کلاینت اجازه دهید اگر نیاز بود پارامتر Duration را انتخاب کند. یا اینکه شاید بخواهید متد بتواند مقادیر Integer یا String را به عنوان پارامتر DisplayValue بپذیرد.

گرچه در VB.NET اجازه دارید پارامترهای اختیاری داشته باشید اما بهترین راه Overloading است. Overloadها متدهای چندگانه هستند. متدهای Overload شده باید امضای متفاوتی داشته باشند. اما نیاز ندارند مقدار بازگشتی و نوع و یا سطح دسترسی مشابهی داشته باشد. وقتی یک متد Overload شده صدا زده می‌شود، CLR نوع آرگومانهای تحویل داده شده در فراخوانی متد را بررسی می‌کند. سپس لیست آرگومان‌ها را با فراخوانی‌ها و امضاهای Overloadهای موجود تطابق می‌دهد.

اگر هیچ Overloadای با نوع آرگومان‌ها تناسب نداشته باشد، یک خطا اعلام می‌شود.

1-7. ساختن متدهای Overload

شما می‌توانید یک متد Overload شده را از راهی شبیه آنچه دیگر متـدها را مـی‌سـازید، درست کنید: بوسیله توصیف متد با یک نام، یک سطح دسترسـی، یـک نـوع بازگـشتی، و یـک لیست آرگومان. یک متد Overload شده باید نامی شبیه متدی موجود اما با امضایی متفاوتی با آن داشته باشد.

سطح دسترسی و نوع بازگشتی می‌تواند مـشابه و یـا متفـاوت باشـد. مثـال زیـر یـک متـد Overload شده را نشان می‌دهد.

```
.This example demonstrates an overloaded method '
(DisplayMessage(ByVal i As Integer Public Sub
(()MessageBox.Show(i.ToString
Sub End
This method has the same name as the previous method, but is '
distinguishable by signature '
(String Public Sub DisplayMessage(ByVal s As
(MessageBox.Show(s
End Sub
```

وقتی یک متد با نام DisplayMessage فراخوانی می‌شود، CLR نوع آرگومان‌های تحویل داده شده را مشخص می‌کند. اگر یک String باشد، متدی که String می‌گیرد، صدا زده می‌شود و اگر یک Integer باشد، متدی که Integer می‌گیرد فراخوانی می‌شود.

8- اینترفیس‌ها

اینترفیس یک قرارداد است. هر شیئ که پیاده سازی می‌شود، برای تامین پیاده سازی اعضای تعیین شده در آن اینترفیس یک ضمانت نامه اینترفیس می‌گیرد.

اگر یک شیئ نیاز به تعامل با اینترفیسی داشته باشد، می‌تواند با هر شیئ که آن اینترفیس را پیاده سازی می‌کند تعامل داشته باشد. یک اینترفیس فقط اعضایی را تعریف می‌کند که بوسیله شیئ که بعدا پیاده سازی خواهد شد ساخته می‌شود.

تعریف اینترفیس هیچ اطلاعاتی درباره پیاده سازی اعضاء جز پارامترهایی که می‌گیرند و نوع مقادیری که آنها برخواهند گرداند، نمی‌دهد و پیاده سازی اینترفیس‌ها کاملا به پیاده سازی کلاس واگذار می‌شود.

بنابراین این امکان دارد که در اشیاء مختلف پیاده سازی‌های مختلفی از اعضای مشابه داشته باشیم. به عنوان مثال، اینترفیسی به نام IShape که یک متد CalculateArea تعریف می‌کند.

کلاس Circle این اینترفیس را برای محاسبه مساحت خود، به طریق متفاوتی با کلاس Square پیاده سازی می‌کند. هر چند که یک شیئ که نیاز به تعامل با IShapeدارد می‌تواند متد CalculateArea را فراخوانی کند و هر دوی Circle یا Square نتایج درستی می‌دهند.

1-8. تعریف اینترفیس‌ها

اینترفیس‌ها با کلمه کلیدی Interface تعریف می‌شوند.

```
Public Interface IDrivable
End Interface
```

این اعلان اینترفیس IDrivable را تعریف می‌کند اما هیچ عضوی برای آن تعریف نمی‌کند. متدهای یک عضو باید با امضای متد تعریف شود، اما بدون بدون تعریف سطح دسترسی مثل Public و Private. سطح دسترسی یک اینترفیسس سطح دسترسی اعضای آن اینترفیس را نیز تعیین می‌کند.

پس اگر یک اینترفیس Public داشته باشید، همه اعضایش هم باید Public باشند. مثال زیر نشان می‌دهد که چطور متدها را به اینترفیس اضافه کنید:

```
Public Interface IDrivable
(Sub GoForward(ByVal Speed As Integer
()Halt Sub
Function DistanceTraveled() As Integer
End Interface
```

همچنین شما می‌توانید خصوصیات را نیز به اینترفیس‌ها اضافه کنید. خصوصیت باید ReadOnly یا WriteOnly تعریف شوند.

مثال زیر را ببینید:

```
Public Interface IDrivable
.This defines a read-only property '
ReadOnly Property FuelLevel() As Integer
End Interface
```

اگرچه شما می‌توانید خصوصیات را در اینترفیس‌ها تعریف کنید، اما نمی‌توانید فیلـدها را در آنها تعریف کنید. این شرط تضمین می‌کند که کلاسهایی که از طریق اینترفیس‌ها تعامل دارند به دادهای درونی یک شیئ دسترسی نداشته باشند.

اینترفیس‌ها همچنین می‌توانند رویدادها را تعریف کنند:

```
Public Interface IDrivable
(System.EventArgs Event OutOfFuel(ByVal sender As Object, e As
End Interface
```

چند شکلی با اینترفیس‌ها (Interfaces Polymorphism with)

هر شیئ که یک اینترفیس خاص را پیاده سازی می‌کند، می‌تواند با هر یک از اشیاء دیگری که به آن اینترفیس نیاز دارند، تعامل کند.

```
(Public Sub GoSomewhere(ByVal v As IDrivable
omitted Implementation '
End Sub
```

این متد نیاز به یک پیاده سازی از اینترفیس IDrivable دارد. هر شیئ که این اینترفیس را پیاده سازی می‌کند می‌تواند به عنوان یک پارامتر به این متد پاس داده شود.

وقتی یک شیئ توسط اینترفیسش تعامل می‌کند، فقط اعضاء آن اینترفیس در دسترس هستند.شما همچنین می‌توانید به طور ضمنی اشیایی که اینترفیس خاصی را پیاده سازی میکنند درست کنید. (توجه کنید در این مثال Truck باید IDrivable را پیاده سازی کند)

```
()Dim myTruck As New Truck
Dim myVehicle As IDrivable
Casts myTruck to the IDrivable interface '
(IDrivable ,myVehicle = CType(myTruck
```

2-8. پیاده سازی اینترفیس‌ها

در VB.Net اینترفیس‌ها بوسیله کلاس‌ها و ساختارها (Structures) با کلمه کلیدی Implements پیاده سازی می‌شوند:

```
Public Class Truck
Implements IDrivable
code omitted Additional implementation '
End Class
```

کلاس‌ها می‌توانند چندین اینترفیس را پیاده سازی کنند. اگر بخواهید کلاسی تعریف کنید که چندین اینترفیس را پیاده سازی کند، می‌توانید نام اینترفیس‌ها را با کاما از هم جدا کنید. مثال زیر را ببینید:

```
Public Class Truck
Implements IDrivable, IFuelBurning, ICargoCarrying
Additional implementation code omitted '
End Class
```

وقتی یک کلاس یا ساختاری یک اینترفیس را پیاده سازی می‌کند، شما باید برای هر یک از اعضای آن اینترفیس پیاده سازی جداگانه‌ای انجام دهید. اگر اینترفیس‌های چندگانه پیاده سازی شوند، باید یک پیاده سازی برای هر عضو هر اینترفیسی انجام دهید.

3-8. پیاده سازی اعضای اینترفیس‌ها

در VB.NET، یک عضو کلاس با کلمه کلیدی Implements یک عضو اینترفیس را پیاده سازی می‌کند. عضو کلاسی که عضو اینترفیس را پیاده سازی می‌کند باید امضایی مشابه آن چه در اینترفیس تعریف شده، داشته باشد. اما نیاز نیست که سطح دسترسی مشابهی با آن داشته باشد. مثال بعد نشان می‌دهد چگونه یک عضو اینترفیس تعریف می‌شود.

```
Public Interface IDrivable
(Sub GoForward(ByVal Speed As Integer
Interface End
Public Class Truck
Implements IDrivable
(Public Sub GoForward(ByVal Speed As Integer
IDrivable.GoForward Implements
Implementation omitted '
End Sub
End Class
```

عضو کلاسی که یک عضو اینترفیس را پیاده سازی می‌کند مجبور نیست که نامی مشابه نام عضو اینترفیس داشته باشد. مثال بعد یک پیاده سازی کاملا معتبر متد GoForward از اینترفیس IDrivable را نشان می‌دهد:

```
(Public Sub Move(ByVal Speed As Integer
IDrivable.GoForward Implements
Implementation omitted '
End Sub
```

هر فراخوانی متد GoForward از اینترفیس IDrivable در مثال قبل، متد Move را از کلاس Truck اجرا می‌کند. همچنین شما می‌توانید سطح دسترسی متفاوتی را برای متد کلاسی که متد اینترفیس را پیاده سازی می‌کند در نظر بگیرید.

به عنوان مثال می‌توانید متـد Public اینتـرفیس را متـد Private کـلاس پیـاده سـازی کنید. اگر این روش را انجام دهید، وقتی کـه دسترسـی درون اینتـرفیس اسـت متـد Public می‌شود، اما وقتی دسترسی به عنوان عضوی از کلاس است Private می‌ماند.

9- خواص و رویدادهای یک فرم در VB

کسانی که با برنامه سازی سروکار دارند حتما اسم بیسیک رو شـنیده‌انـد بیـسیک بهتـرین زبان برنامه نویسی برای مبتدیان به شمار می رود ولی هرگز نبایید به این زبان اکتفـاکرد چـون از سایر زبان ها مانند پاسکال ضعیف تراست .

برای برنامه نویس های که برنامه‌های بزرگ می نویسن بعضی از کارها مانند درسـت کـردن منو ها ومدیریت منوها ویا درست کردن یک فرم ساده هم بسیار مشکل و وقـت گیـر اسـت بـه همین دلیل به فکر درست کردن برنامه‌های که دارای فرم های آماده وقسمت های که مدیریت آنها بسیار آسان تر از سایر محیط های برنامه سازی گذشته است افتادند یکی از این زبان هـای برنامه نویسی ویژوال بیسیک است کار با آن بسیار راحت وشیرین است برنامه‌ای که بـا ویـژوال بیسیک نوشته می‌شود بسیار زیبا وبرای کاربر هم کار با آن ساده است .

9-1. فرم :

یک پروژه می تواند شامل یک یا چند فرم باشد که می توان در داخـل هـر فـرم، اشـیایی را اضافه نمود و خصوصیاتی را برای هر کدام از آنها به دلخواه تعیین کرد و برای هر رویـدادی کـه می تواند برای هر شی یا فرم خواصی اتفاق بیفتد و برای ما اهمیـت دارد، رفتارهـایی را تعیـین کرد.

9-2. رویدادهای یک فرم :

هر عنصر در ویژوال بیسیک،تعدادی رویداد مربوط به خود را دارد یعنی ما تعیین می کنـیم که در مقابل هر یک از این رویدادها چه عکس العملی را انجام انجام دهد که ایـن رویـدادها در بعضی از عناصر مشترک است. در زیر به معرفی بعضی از این رویدادها می پردازیم:

رویدادClick:هنگامی که کاربر روی یکی از عناصر برنامه یا خود فرم کلیک کند.

رویداد DbClick: هنگامی که کاربر روی یکی از عناصر برنامه یا خود فرم دو بار کلیک کند.

رویدادActivate:زمانی که فرم روی صفحه نمایش ظاهر می‌شود.

رویدادDeactivate:هنگامی که کاربر روی یه برنامه دیگر برود ویا فرم از روی صفحه نمایش محو شود.

رویدادDragDrop:زمانی که عنصر بر روی فرم (به وسیله کشیدن موس) قرار گیرد.

رویدادDragOver:زمانی که عنصر روی فرم حرکت کند.

رویدادGotFocus:زمانی که مکان نما بر روی عنصری قرار گیرد.

رویدادInitialize:هنگامی که فرم ایجاد می‌شود.

رویدادKeyDown:هنگامی که کاربر یکی از کلیدهای صفحه کلید (به جز Alt,Shift,Inser) پایین نگه دارد.

رویدادKeyPass:زمانی که کلیدی فشرده شود. (به جز Alt,Shift,Inser)

رویدادKeyUp:این رویداد هنگام رها کردن کلید فشرده شده فراخوانی می‌شود.

رویدادLoad:در زمان ایجاد فرم این رویداد فراخوانی می‌شود.

رویدادLostFocus:هنگامی که مکان نما از روی عنصری خارج می‌شود.

رویدادMouseDown:وقتی که یکی از کلیدهای موس پایین نگه داشته شود.

رویدادMouseMove:وقتی که اشاره گر موس روی فرم یا عنصری حرکت داده شود.

رویدادMouseUp:زمانی که کاربر کلید فشرده شده موس را رها کند.

رویدادPaint:این رویداد زمان بازسازی مجدد موس فراخوانی می‌شود.

رویدادQueryUnLoad:زمانی که کاربر سعی می کند فرم را ببندد.

رویدادResize:زمانی که کاربر اندازه فرم را تغییر دهد.

رویدادTerminate:هنگامی که کاربا فرم خاتمه می یابد.

رویدادUnLoad:این رویداد زمان حذف فرم فراخوانی می‌شود.

در ویژوال بیسیک هر عنصرویژوال خواص مربوط به خود را دارد.از جمله این خواص می توان به خواصی مانند اندازه، شکل،فونت،رنگ وسایر خواص رانام برد هر عنصر خواص خاص خودش را دارد ولی بعضی ازخواص در تمام عناصر مشترک است می توان خواص عنصررابه صورت روبه رو نوشت مقدار=خاصیت.نام عنصر مانند " Text1.Text="Danesh va computer حال به معرفی خواص فرم می پردازیم

خاصیت Name:این خاصیت، برای تعیین نام عنصر(فرم)درویژوال بیسیک بکار میرود.

خاصیت Backcolor:این خاصیت، برای تعیین رنگ زمینه فرم بکار میرود.

خاصیت Borderstyle:این خاصیت، نوع حاشیه فرم را مشخص می کند.

خاصیت Caption:این خاصیت، برلی تعیین عنوان فرم بکار میرود.

خاصیت Enable:این خاصیت، برای فعال یا غیر فعال کردن فرم بکار می رود.

خاصیت Font:این خاصیت، برای مشخص کردن فونت نمایش اطلاعات بکار میرود.

خاصیتVisible:این خاصیت ،تعیین می‌کند که فرم قابل رویت باشد یا نه.

خاصیت Height:این خاصیت، ارتفاع فرم را مشخص می کند.

خاصیتTop:این خاصیت ،فاصله کناره بالای پنجره‌ای فرم از حاشیه بالای صفحه نمایش را مشخص می کند.

خاصیتWidth:این خاصیت،عرض فرم را مشخص می کند.

خاصیت Left:این خاصیت ،برای تعیین فاصله فرم با لبه سمت چپ صفحه نمایش استفاده می‌شود .

خاصیت ControlBox:این خاصیت تعیین می کند که کادر کنترل پنجره فرم (دکمه‌های حداقل،حداکثر،منوی فرم ،اندازه فرم) نمایش داده شود یا خیر.(بر اساس Trueو False)

خاصیت Icon:زمانی که فرم به حداقل تغییر یابد،فرم به شکل یک آیکن دیده می‌شود .(با انتخاب ایکون از درون هارد خود بر روی نوار وظیفه ویندوز ظاهر می‌شود)

خاصیت MaxButton:برای نمایش دادن یا ندادن دکمه حداکثر اندازه پنجره است.

خاصیت MinButton: برای نمایش دادن یا ندادن دکمه حداقل کردن پنجره است.

خاصیت MoveAble:این خاصیت تعیین می کند که کاربر بتوانند پنجره را حرکت دهدیا خیر(True,False)

خاصیت WindowsState:این خاصیت،وضعیت فرم را مشخص می کند. (۱--Normal,2-Max,3-Min)

خاصیت MousePointer:این خاصیت ،برای تعیین شکل موس بکار می رود

خاصیتToolTipText:چانچه مکان نما بر روی عنصری قرار گیرد ،اطلاعات کمکی مربوط به آن شکل ظاهر می‌شود.

خاصیتDrawWidth:این خاصیت برای تعیین عرض خطوطی که بر روی فرم رسم می‌شوند، به کار می رود.

خاصیت AutoRedraw:اگر مقدار این خاصیت Trueباشد،ویژوال بیسیک یک کپی از اطلاعات فرم نگهداری می کندتادر صورت لزوم آن را دوباره رسم کند.

خاصیتDrawStyle:این خاصیت،برای تعیین سبک قلم گرافیکی به کار می رود.

خاصیت DrawMode:این خاصیت ،نحوه ترکیب قلم گرافیکی راباعناصر روی فرم تعیین می‌کند.

خاصیتFillColor: این خاصیت ،رنگی را تعیین می کند که شکل گرافیکی روی فرم تعیین می کند.

خاصیتFillStyle: این خاصیت حاشیه‌ای را تعیین می کند که شکل گرافیکی با ید با آن پر شود.

خاصیتScaleWidth:این خاصیت پهنای بخش کاری فرم را مشخص می کند.

خاصیت ScaleHight: این خاصیت برای تعیین ارتفاع بخش کاری فرم بکار می رود.

خاصیتForeColor: این خاصیت برای تعیین رنگ قلم بکار می رود.

خاصیتAppearance:این خاصیت تعیین می کند که فرم دارای نمای سه بعدی باشد یا نه.

خاصیت Tag:این خاصیت ،یک عدد صحیح است که اطلاعاتی راجب به فرم در خود نگهداری می کند.

فصل چهارم

1- دستورات بکار برده شده در سایت اختصاصی بیمارستان مهر

لینکهایی که مشاهده می فرمایید به ترتیب :

- **صفحه اصلی**
- **مدیریت**
- **جستجو**
- **ارتباط با بیمارستان**
- **درباره بیمارستان**
- **لیست کارمندان بیمارستان**
- **لیست دکتر های بیمارستان**
- **لیست پرستاران بیمارستان**
- **لیست بیماران بیمارستان**
- **جدید ترین تحقیقات بیمارستان**

می باشد.

1-1. لینک صفحه اصلی :

با زدن این لینک صفحه اصلی یا فرم Default.aspx نمایش داده می شود ، دستوری که برای نشان دادن این فرم بکار میرود به شرح زیر می باشد :

```
<a href = "Default.aspx" ></a>
```

البته قابل به ذکر است که این دستور یک دستور یا تگ HTML می باشد

1-2. لینک مدیریت :

از آنجایی که این سایت یک سایت پویا یا دینامیک می باشد ، مدیر سایت باید امکان حذف وویرایش و حتی افزودن اطلاعات را در سایت داشته باشد. با زدن این لینک صفحه یا فرم login.aspx نمایش داده می شود .

در صفحه login.aspx یک فرم وجود دارد که از کاربر یا مدیر سایت نام کاربری و کلمه عبور را می خواهد ، اگر کلمه عبور و نام کاربری درست وارد شود کاربر از این صفحه به صفحه main.aspx لینک می شود و اگر نام کاربری و کلمه عبور صحیح وارد نشود در همان صفحه یعنی صفحه login.aspx میماند و یک هشدار به کاربر نشان داده میشود که نام کاربری و کلمه عبور اشتباه وارد شده است.

دستوراتی که در این صفحه بکار برده شده به شرح زیر می باشد :

```
  Dim strname As String
        Dim strpass As String

        strname = txtuname.Text.Trim
        strpass = txtpass.Text.Trim

        If strname = "" Then
            MsgBox("نمایید وارد را کاربری نام لطفا")
        Else
            If strpass = "" Then
                MsgBox("نمایید وارد را عبور کلمه لطفا")
            Else

                Dim strconn As String
                Dim objconn As SqlConnection = Nothing
                strconn = "Data Source=.;Initial Catalog=Hospital;Integrated Security=True"
                objconn = New SqlConnection(strconn)
                objconn.Open()
```

```
                Dim strsql As String
                Dim objcommand As SqlCommand =
Nothing

                strsql   =   "select   *   from
Tbl_admin where [Uname] = @uname and [Pass] =
@pass"
                objcommand = New SqlCommand
                objcommand.CommandText = strsql
                objcommand.CommandType        =
CommandType.Text
                objcommand.Connection = objconn

objcommand.Parameters.Add("@uname",
SqlDbType.NVarChar, 50)

objcommand.Parameters.Add("@pass",
SqlDbType.NVarChar, 50)

objcommand.Parameters("@uname").Value = strname

objcommand.Parameters("@pass").Value = strpass

                Dim dr As SqlDataReader
                dr = objcommand.ExecuteReader



                If dr.Read() Then
```

```
                    If    dr.Item("Uname")    =
strname And dr.Item("Pass") = strpass Then
                        Session("login") = 1

Response.Redirect("main.aspx")
                    Else
                        Session("login") = 2
                        MsgBox("عبـور کلمـه و کاربری نام
نیست معتبر")


                    End If
                Else
                    MsgBox("معتبـر عبـور کلمه و کاربری نام
نیست")
                End If


            End If
        End If
```

همان طور که در کد بالا مشاهده می فرمایید در ابتدا متغیـر هـا را تعریـف مـی کنـیم.کـه ساختار این عمل به شرح زیر می باشد :

Dim نام متغیر As نوع متغیر

سپس می بایست به متغیر های تعریف شده مقدار بدهیم که به شرح زیر می باشد :

نام متغیر = مقدار

سپس چک میکنیم که آن دو Text Field نام کاربری و کلمه عبور خالی نباشند ، کـه در اینجا از دستور if استفاده کرده ایم که ساختار این دستور به شرح زیر می باشد :

If شرط Then

دستورات اجرایی

End if

ما بین دستور If از دستو Msgbox استفاده کرده ایم که ساختار آن به شرح زیر می باشد :

Msgbox("جمله ای که در هشدار نمایش داده می شود")

حال اگر آن Text Field ها خالی نباشند زمان آن رسیده تا رشته های داخـل آنهـا را بـا جدول Tbl_admin داخل بانک اطلاعاتی مقایسه کنیم که به زبان عامیانه به شرح زیر است : اگر نام کاربری برابر با فیلد Uname داخل بانک اطلاعاتی بود و کلمه عبور هم برابـر بـا فیلـد Pass داخل بانک اطلاعاتی بود لینک شو به صفحه main.aspx و همراه با لینک شـدن یـک

Session هم با نام login بساز و مقدار آن را هم برابر با ۱ قـرار بـده ، در غیـر ایـن صـورت لینک شو به صفحه login.aspx و مقدار Session ، login را برابـر بـا ۲ قـرار بـده و یـک هشدار هم نمایش بده.

که همانطور که مشاهده می فرمایید دستورات آن به شرح زیر می باشد :

```
                Dim strconn As String
                Dim objconn As SqlConnection =
Nothing
                strconn          =          "Data
Source=.;Initial    Catalog=Hospital;Integrated
Security=True"
                objconn          =          New
SqlConnection(strconn)
                objconn.Open()


                Dim strsql As String
                Dim objcommand As SqlCommand =
Nothing

                strsql   =   "select   *   from
Tbl_admin where [Uname] = @uname and [Pass] =
@pass"
                objcommand = New SqlCommand
                objcommand.CommandText = strsql
                objcommand.CommandType         =
CommandType.Text
```

```
                objcommand.Connection = objconn

objcommand.Parameters.Add("@uname",
SqlDbType.NVarChar, 50)

objcommand.Parameters.Add("@pass",
SqlDbType.NVarChar, 50)

objcommand.Parameters("@uname").Value = strname

objcommand.Parameters("@pass").Value = strpass

                Dim dr As SqlDataReader
                dr = objcommand.ExecuteReader

                If dr.Read() Then

                    If    dr.Item("Uname")    =
strname And dr.Item("Pass") = strpass Then
                        Session("login") = 1

Response.Redirect("main.aspx")
                    Else
                        Session("login") = 2
                        MsgBox("عبـور کلمـه و کاربری نام
نیست معتبر")
```

```
                End If
            Else
                MsgBox("نام کاربری و کلمه عبـور معتبـر
نیست")
            End If
        End If
    End If
```

3-1. لینک جستجو :

بـا زدن ایـن لینـک صـفحه یـا فـرم Search.aspx نمـایش داده مـی شـود. در صـفحه Search.aspx هم سه لینک وجود دارد که به شرح زیر می باشد :

- جستجو در میان دکترها
- جستجو در میان پرستاران
- جستجو در میان بیماران

جستجو در میان دکترها به صفحه searchdoctor.aspx لینک شده است .

جستجو در میان پرستاران به صفحه searchnurse.aspx لینک شده است .

جستجو در میان بیماران هم به صفحه searchpatient.aspx لینک شده است

Searchdoctor.aspx :

در این صفحه می بایست رشته ای را که کاربر وارد می نماید مورد جستجو قرار گیرد از جدول Tbl_doctors از بانک اطلاعاتی و سپس اگر موردی یافت شد نمایش داده شود.

که این عملیات به شرح زیر می باشد :

در ابتدا می بایست یک Text Field به داخل فرم بیندازیم که برای این کار کافی است از داخل toolbox سمت چپ صفحه یک Text Field را انتخاب کنیم و به داخل صفحه بکشیم .

سپس باید یک نام منحصر به فرد برای این Text Field انتخاب تماییم. که ما در اینجا نامش را txttakhasos قرار داده ایم تا در ادامه برنامه بتوانبم از این نام استفاده نماییم.

در قسمت بعدی می بایست یک Sql Data Source وارد صفحه نماییم . برای این کار هم کافی است از toolbox کنار صفحه از پنل Data ، Sql Data Source را انتخاب نماییم و به داخل صفحه بکشیم . این sql data source را باید به جدول Tbl_doctors نسبت بدهیم .

پس از این کار باید یک Grid view به داخل صفحه بیاوریم برای نمایش اطلا عاتی که Sql Data Source از جدول داخل بانک اطلاعاتی می خواند .

برای آنکه Sql Data Source تنها آن اطلا عاتی را بیاورد که کاربر در آن Text Field بالا وارد نموده می بایست برای آن یک شرط یا Where نوشت که به شرح زیر می باشد :

Select * from Tbl_doctors where [Takhasos] = txttakhasos

در ما بقی صفحات جستجو هم این چنین برتامه ای نوشته شده است .

4-1. لینک ارتباط با بیمارستان :

با زدن این لینک صفحه contact.aspx نمایش داده می شود . که در این صفحه شماره تلفن ، شماره فکس ، ایمیل بیمارستان برای ارتباط با بیمارستان نشان داده می شود .

5-1. لینک درباره بیمارستان:

که در این صفحه خلاصه از بیو گرافی بیمارستان نشان داده شده است

6-1. لینک لیست کارمندان بیمارستان :

با زدن این لینک صفحه employee.aspx نمایش داده می شود .

در این صفحه اطلاعات از جدول Tbl_employee داخل بانک اطلاعاتی خوانده می شود و به کاربران نشان داده می شود.

دستورات بکار برده شده برای این صفحه به شرح زیر می باشد:

ابتدا می بایست همانند صفحات جستجو یک Sql Data Source به داخل صفحه بیندازیم . که این Sql Dta Source تمام اطلاعات را از جدول Tbl_employee میخواند و در خود نگه می دارد .که دستو ر آن به شرح زیر می باشد :

```
Select * from Tbl_employee
```

حال می بایست اطلاعاتی را که درون Sql Data Source می باشد را نمایش دهیم.

برای این کار به یک Grid view احتیاج داریم.

که درست همانند صفحات جستجو که در قسمتهای بالا توضیح داده شـد Grid view را به داخل صفحه وارد می کنیم و اطلاعاتی را که در داخل Sql Data Source می باشد را بـه این Grid view ، Bound می کنید

7-1. لینک لیست دکترهای بیمارستان :

این لینک هم درست همانند لینک لیست کارمندان بیمارستان می باشد با این تفاوت که با زدن این لینک صفحه doctor.aspx نمایش داده می شود .

8-1 . لینک لیست پرستاران بیمارستان :

این لینک هم درست همانند لینک لیست کارمندان بیمارستان می باشد با این تفاوت که با زدن این لینک صفحه nurse.aspx نمایش داده می شود .

9-1. لینک لیست بیماران بیمارستان :

این لینک هم درست همانند لینک لیست کارمندان بیمارستان می باشد با این تفاوت که با زدن این لینک صفحه patient.aspx نمایش داده می شود.

10-1. لینک جدید ترین تحقیقات بیمارستان :

با زدن این لینک صفحه tahghigh.aspx نمـایش داده مـی شـود ، در ایـن صـفحه هـم همانند قسمتهای قبلی سایت احتیاج به یک Sql Data Source و یک Grid view بـرای نمایش اطلاعات داریم.

که این اطلاعات را از جدول Tbl_tahghigh از داخل بانک اطلاعاتی می خواند و نمـایش می‌دهد .

نتیجه گیری

در کل نتیجه ی که ما با راه اندازی این سایت میگیریم این است که در صـورت استفاده بیمارستان از این سایت کمک شایانی به پزشکان و پرستاران و همه کسانی که به نوعی از بیماران نگه داری میکنند میشود.و بستگان و نزدیکان بیماران چه در داخل و چه در خارج از کشور میتوانند با استفاده از ایـن سـایت از وضـعیت بیمـار خود اطلاع حاصل کنند.

منابع

www.p30world.com

www.webgostarco.com

www.srco.ir

www.remisco.ir

www.sina.sharafi.ir

www.rasamoj.com

www.old.tebyan.net

www.aftab.ir

www.srco.ir//Articles/DocView.asp?ID=210

www.4guysfromrolla.com

www.srco.ir//Articles/DocView.asp?ID=210