بسم الله الرحمن الرحیم

مقالات

VB.NET

انواع داده ای در Visual Basic.NET

برای تعریف یک متغیر در ویژوال بیسیک دات نت از عبارت Dim استفاده می شود. برای مثال چمله زیر یک متغیر جدید به نام myVar تعریف می کند:

Dim myVar

هنگامی که یک متغیر مانند فوق بدون در نظر گرفتن نوع آن تعریف شود، آن متغیر بعنوان یک شئ (Object) در نظر گرفته می شود. یک متغیر از نوع شئ در دات نت می تواند شامل انواع داده از قبیل رشته، عدد یا انواع پیچیده تر مانند کلاس باشد.

اغلب نیازی نیست که از متغیرهای از نوع شئ در برنامه استفاده شود چرا که این متغیرها باعث کاهش کارآیی و سرعت برنامه می شود. هنگامی که از یک متغیر از نوع شئ استفاده شود، نوع مناسب آن متغیر در زمان اجرا تشخیص داده می شود و این باعث کندی می شود. بنابراین بهتر است که نوع داده را از ابتدا مشخص کنیم.

جهت مشخص نمودن نوع داده از کلمه کلیدی As استفاده می کنیم. بعنوان مثال یک متغیر از نوع رشته بصورت زیر تعریف می شود:

Dim myVar As String
مجموعه دات نت انواع داده ای زیر را پشتیبانی می کند:

- Boolean: جهت مقادیر درست یا نادرست
- Byte: جهت مقادیر صحیح صفر تا ۲۵۵ (تنها مقادیر مثبت)
- Char: جهت مقادیر نویسه ها یا حروف یونیکد
- Date: جهت مقادیر تاریخ و زمان بین اول ژانویه ۰۰۰۱ و ۳۱ دسامبر ۹۹۹۹
- Decimal: جهت مقادیر صحیح بین صفر و مثبت و منفی
۷۹,۲۲۸,۱۶۲,۵۱۴,۲۶۴,۳۳۷,۵۹۳,۵۴۳,۹۵۰,۳۳۵ (۱۶ بایت)
- Double: جهت مقادیر اعشاری با دقت مضاعف
- Integre: جهت مقادیر صحیح بین ۲,۱۴۷,۴۸۳,۶۴۷+ و ۲,۱۴۷,۴۸۳,۶۴۸- (۴ بایت)
- Long: جهت مقادیر صحیح بین ۹,۲۲۳,۳۷۲,۰۳۶,۸۵۴,۷۷۵,۸۰۸- و
۹,۲۲۳,۳۷۲,۰۳۶,۸۵۴,۷۷۵,۸۰۷+ (۸ بایت)
- Short: جهت مقادیر صحیح بین ۳۲,۷۶۸- و ۳۲,۷۶۷+ (۲ بایت)
- Single: جهت مقادیر اعشاری با دقت واحد
- String: جهت مقادیر رشته ای با تعداد حروف صفر تا دو میلیارد حرف

پر کاربردترین انواع داده ای عبارتند از Integer، Decimal، Date، Boolean و String.

دقت کنید که نوع داده ای Currency یا Money برای مقادیر پولی در نظر گرفته نشده است و شما می توانید برای مقادیر پولی از نوع Decimal استفاده کنید.

ساختارهای شرطی در Visual Basic.NET

این گفتار برآنیم تا ساختارهای شرطی در ویژوال بیسیک دات نت را بررسی کنیم. این ساختارها If..Then و Select..Case می باشند.

ساختار If..Then
ابتدائی ترین ساختار شرطی در ویژوال بیسیک دات نت ساختار If..Then می باشد. با استفاده از این ساختار هنگامی که شرط مورد نظر برقرار باشد، می توان دستور یا دستورات متعددی را اجرا نمود.

بعنوان مثال در برنامه زیر اگر زمان سیستم بعد از ظهر را نشان دهد، جمله " Good Evening IranASP.NET" بر روی صفحه نمایش داده می شود.

```
<%
Dim myTime As DateTime
myTime = Now

if Hour(myTime) >= 12 then
Response.write ("Good Evening IranASP.NET !")
end if
%>
```

همچنین ساختار If..Then عبارت Else را هم پشتیبانی می کند. اگر شرط مربوط به If برقرار نباشد، دستورات موجود در قسمت Else اجرا می شوند. به مثال زیر توجه فرمائید.

```
<%
Dim myTime As DateTime
myTime = Now

if Hour(myTime) >= 12 then
Response.write ("Good Evening IranASP.NET !")
else
Response.write ("Good Morning IranASP.NET !")
end if

%>
```

ساختار Select..Case

در ساختار Select..Case می توان مقداری را با مقادیر مختلفی مقایسه کرده و دستورات مربوط به مقدار یافت شده را اجرا نمود. بعنوان مثال قطعه برنامه زیر پیامهای مختلفی را برحسب نوع مرورگر نمایش می دهد.

```
<%
Dim strBrowser As String

strBrowser = Request.Browser.Browser
Select Case strBrowser
Case "IE"
```

```
Response.Write( "You are using Internet Explorer!" )
Case "Netscape"
Response.WRite( "You are using Netscape!" )
Case Else
Response.Write( "What browser are you using?" )
End Select
%>
```

دقت داشته باشید که ساختار Select..Case در برنامه فوق دارای یک قسمت Case Else می باشد. هرگاه هیچ یک از حالات مقایسه ای برقرار نبود، دستورات موجود در قسمت Case Else اجرا می گردند. استفاده از Case Else اختیاری است.

متغیرهای ایستا در ASP.NET

در ASP همواره از شئ Application برای ذخیره متغیرهای سراسری استفاده می شد. این عمل از لحاظ اختصاص فضای حافظه چندان مناسب نبود. در دات نت می توانیم با سود بردن از خواص متغیرهای ایستا در اکثر موارد نتیجه بهتری بدست آوریم. این روش در اکثر موارد سریعتر از استفاده از شئ Application خواهد بود.

در دات نت اکثر اشیاء به صورت کلاس در نظر گرفته می شوند که فایل global.asax نیز از این قائده پیروی می کند. برای استفاده از این روش در ابتدا باید به این فایل نام یک کلا س را اختصاص دهیم. دقت کنید که همیشه سعی می کنیم در نامگذاری از اسامی که راهنمای ما باشند استفاده کنیم. برای مثال در اینجا از نام myglobal استفاده می کنیم. به منظور انجام این کار از کد زیر استفاده می شود.

```
<%@ Application Classname="MyGlobals" %>
```

سپس با استفاده از تگ Script متغیرهای خود را تعریف می نماییم. دقت کنید که باید از کلمات کلیدی Public و Shared هم استفاده نما یید.

```
<Script language="vb" runat="server">
  Public Shared sAli as String = "This is just a test"
</Script>
```

با کد بالا ما متغیر خود را به نام sAli تعریف نمودیم. حال با استفاده از نام کلاس و این نام می توا نیم آن را در تمام صفحات خود به صورت مستقیم صدا کنیم.

```
x = MyGlobals.sAli
```

کدهای نمونه را می توانید مشاهده کنید.

```
<%@ Application Classname="MyGlobals" %>
<Script language="vb" runat="server">
  Public Shared sGreeting as String = "This is just a test"
</Script>
<% @Page Language="VB" %>
```

```
<HTML>
   <HEAD>
      <script Language='vb' runat=server>
         Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
            Label1.Text = MyGlobals.sGreeting
         End Sub
      </script>
   </HEAD>
   <body>
      <asp:label runat=server id=Label1></asp:label>
   </body>
</HTML>
```

آشنایی با Visual Basic.NET– متغیرها

متغیرها حاملهای دربرگیرنده مقادیری هستند که در جریان یک برنامه کاربردی می توانند تغییر کنند. برنامه نویسی بدون آنها، اگر نگوئیم غیرممکن، اما به سختی امکان پذیر است. در ASP.NET، متغیرها در چند لایه وجود دارند.

لایه اول، لایه Application است. در اینجا در هر صفحه برنامه کاربردی، همه متغیرها در دسترس همه کاربران قرار دارد. معمولا یک قطعه داده نسبتا کوچک که در معرض استفاده مکرر است، همانند اطلاعات مربوط به ارتباط با بانک اطلاعاتی، در این لایه ذخیره می شود.

لایه دوم، لایه Session است. در این لایه، همه متغیرها در سرتاسر برنامه کاربردی یا تا برقرار بودن جلسه کاربر، در دسترس یک کاربر بخصوص هستند.

لایه سوم و آخر، لایه Page است. در اینجا همه متغیرهای تعریف شده در صفحه در سرتاسر آن صفحه در دسترس هستند. متغیرهای تعریف شده در یک صفحه ASP.NET از صفحه دیگر قابل دستیابی نیستند.

به علاوه در زمان استفاده از code-behindها در صفحات ASP.NET، در پیمانه های کلاس خود به متغیرهای سطح Procedure، Public،Private و Block دسترسی دارید. متغیرهای عمومی در کلاس هایی که در آنها ایجاد شده اند و نیز پیمانه های کلاس دیگر قابل دسترسی هستند. جلوی این متغیرها کلمه کلیدی Public قرار می گیرد. متغیرهای خصوصی تنها در دسترس توابع و روال های کلاس هایی هستند که در آنها تعریف شده اند.

قبل از متغیرهای خصوصی کلمه کلیدی Private قرار می گیرد. متغیرهای سطح Block تنها در بلوک های برنامه ایجاد کننده خود در دسترس می باشند. قبل از متغیرهای سطح Procedure و Block کلمه کلیدی Dim قرار می گیرد.

نکته حائز اهمیت این است که باید به خاطر بسپاریم که همیشه باید نوع داده های متغیر خود را تعریف کنیم. در زیر لیستی از انواع داده های پشتیبانی شده در VB.NET آورده شده است.

Boolean: درست یا نادرست
Byte: برای داده های عددی ۰ تا ۲۵۵
Char: برای یک کاراکتر یونیکد

Date: اطلاعات تاریخ و زمان
Decimal: اطلاعات عددی صحیح
Double: داده های عددی اعشاری
Integer, Long, Short, Single: داده های عددی صحیح
Object: متغیر حامل پیش فرض جهت نگهداری هر نوع شئ
String: رشته های حرفي

در نگارشهای قبلی ویژوال بیسیک و در VBScript همه متغیرها بطور پیش فرض از نوع داده variant بودند. Variant در VB.NET وجود ندارد و با نوع داده Object جایگزین شده است.

بررسی اعتبار داده ها توسط VB.NET در فرم های وب

فریم ورک دات نت ، شامل مجموعه ای از کنترل های لازم بمنظور بررسی اعتبار و صحت داده های ورودی است که با استفاده از آنان در فرم های وب و ASP.NET ، می توان داده ورودی توسط کاربر در هر یک از فیلدهای موجود بر روی یک فرم وب را بررسی و در صورت عدم رعایت شرایط لازم ، پیام خطاء مناسبی را ارائه نمود. در این مقاله به تشریح نحوه استفاده از کنترل های صحت داده ، خواهیم پرداخت .

کنترل های سرویس دهنده اعتبار داده

کنترل های سرویس دهنده اعتبار داده ، امکان بررسی صحت داده مرتبط با یک کنترل سرویس دهنده ورودی نظیر یک TextBox را فراهم و در صورتیکه ماحصل بررسی بعمل آمده ، مثبت نباشد ، یک پیام خطاء نمایش داده خواهد شد. هر یک از کنترل های اعتبار داده ، عملیات خاصی را در ارتباط با بررسی صحت داده ها انجام خواهند داد . مثلا" می توان با استفاده از CompareValidator ، عملیات بررسی صحت داده ها را در ارتباط با یک مقدار خاص انجام و یا با استفاده از کنترل RangeValidator یک محدوده قابل قبول از مقادیر مورد نظر در ارتباط با یک فیلد ورود اطلاعات را انجام داد. در چنین مواردی ، حتی می توان با استفاددە از کنترل CustomValidator ، شرایط خاص مورد نظر خود برای بررسی صحت داده ، تعریف و ایجاد نمود. با توجه به اینکه ، پیام خطاء در کنترل اعتبار داده نمایش داده می شود، می توان مکان مورد نظر برای نمایش پیام خطاء را نیز مشخص کرد . در این راستا ، همچنین می توان با استفاده از کنترل ValidationSummery ، خلاصه ای از نتایج تمامی کنترل های بررسی صحت داده ها را نمایش داد .
عملیات بررسی صحت داده در یک صفحه زمانی محقق خواهد شد که یک کنترل Button نظیر Button,ImageButton و یا LinkButton کلیک گردد . در این رابطه می توان با مقدار دهی False به خصلت CausesValidation ، باعث ممانعت در رابطه با بررسی صحت داده گردید . روش فوق ، در مواردی نظیر فشردن یک دکمه Cancel و یا Clear استفاده می گردد (در چنین مواردی عملا" ما قصد بررسی صحت داده را نخواهیم داشت) . مقدار خصلت فوق ، بصورت پیش فرض در ارتباط با یک Button از نوع Cancel و یا Clear مقدار False در نظر گرفته خواهد شد .
در نمونه مثالی که در ادامه تشریح و به بررسی آن خواهیم پرداخت از چهار کنترل اعتبار داده استفاده شده است که لازم است در ابتدا مختصرا" با عملکرد هر یک از آنان آشنا شویم .

* RequiredFieldValidator . کنترل فوق ، مقدار یک کنترل ورودی را بمنظور حصول اطمینان از درج داده در فیلد مورد نظر توسط کاربر ، بررسی می نماید . در صورتیکه مقدار مشخص شده در فیلد مورد نظر نسبت به مقدار اولیه خود تغییر نکرده باشد ،بمنزله عدم صحت و اعتبار داده بوده و نشان دهنده این موضوع است که کاربر در فیلد مربوطه داده مورد نظر را وارد ننموده است . در حقیقت در چنین شرایطی الزامی برای درج داده در ارتباط با یک کنترل ورودی بوجود آمده و کاربر

نمی تواند فیلد مربوطه را بدون اعمال تغییرات (درج داده) رها نموده و نسبت به آن بی تفاوت باشد . مقدار پیش فرض، یک رشته خالی ("") بوده و نشاندهنده این واقعیت است که می بایست مقداری در آن درج تا در آزمون بررسی صحت داده موفقیت حاصل گردد . در این رابطه فضاء خالی اضافه شده در ابتدا و انتهای کنترل ورودی حذف و در ادامه عملیات بررسی صحت داده انجام خواهد شد . بدین ترتیب از درج فضای خالی در کنترل ورودی برای بررسی صحت داده ممانعت بعمل می آید . در برخی موارد لازم است که در مقابل مقدار پیش فرض (رشته خالی) برای یک کنترل ورودی از مقدار اولیه مورد نظر خود استفاده نمائیم (با توجه به سیاست های طراحی بخش رابط کاربر) . استفاده از روش فوق ، در مواردیکه دارای یک مقدار پیش فرض در ارتباط با یک کنترل ورودی بوده و قصد داریم که کاربر را ملزم به انتخاب یک مقدار دیگر نمائیم ، توصیه می گردد . مثلا" می توان از یک کنترل ListBox بهمراه یک Entry انتخاب شده بصورت پیش فرض که شامل دستورالعمل هائی برای کاربر بمنظور انتخاب یک آیتم از لیست است ، استفاده نمود . در چنین مواردی لازم است که کاربر یک آیتم دیگر موجود در لیست را از طریق کنترل انتخاب و در این رابطه نمی بایست کاربر آیتمی را انتخاب نماید که شامل دستورالعمل است . بدین ترتیب در صورتیکه کاربر همان مقدار اولیه پیش فرض را انتخاب نماید ، کنترل RequiredFieldValidator ، پیام خطاء مورد نظر خود را نمایش خواهد داد . بمنظور مشخص نمودن مقدار اولیه یک کنترل ورودی، می توان از خصلت InitialValue استفاده کرد.
با توجه به سیاست های طراحی رابط کاربر ، می توان از چندین Validator در ارتباط با یک کنترل ورودی مشابه و یکسان استفاده نمود. مثلا" می توان از یک کنترل RequiredFieldValidator ، بمنظور حصول اطمینان از درج داده در کنترل مورد نظر استفاده و در همان حال از کنترل دیگری نظیر RangeValidator نیز استفاده تا این اطمینان ایجاد گردد که ورودی درج شده در کنترل مورد نظر در محدوده قابل قبول و مجاز می باشد .

* کنترل ValidationSummary . کنترل فوق ، امکان ارائه اطلاعات مختصر در رابطه با تمامی کنترل های بررسی صحت داده موجود در یک صفحه وب و در یک موقعیت (مکان) را فراهم می نماید. خلاصه اطلاعات موردنظر را می توان با استفاده از روش های متفاوتی نظیر : یک پاراگراف ، یک لیست و یا یک Bulleted list . ارائه نمود. نحوه نمایش اطلاعات با استفاده از خصلت DisplayMode مشخص خواهد شد .پیام خطاء نمایش داده شده در کنترل ValidationSummary برای هر یک از کنترل های بررسی صحت داده موجود در صفحه ، توسط خصلت ErrorMessage مربوط به هر یک از کنترل ها ، مشخص می گردد . در صورتیکه خصلت ErrorMessage مربوط به کنترل بررسی صحت داده ، مقداردهی نگردد، هیچگونه پیام خطائی در ارتباط با آن کنترل خاص توسط کنترل ValidationSummary نمایش داده نخواهد شد. برای مشخص نمودن عنوان گزارش خطاء، می توان از خصلت HeaderText استفاده کرد. بمنظور کنترل نمایش اطلاعات کنترل ValidationSummary ، می توان از خصلت ShowSummary استفاده نمود . برای نمایش خلاصه اطلاعات مورد نظر در یک Message Box می توان مقدار خصلت ShowMessageBox را True در نظر گرفت .

* کنترل RegularExpressinoValidator . کنترل فوق ، بمنظور تطبیق داده ورودی در یک کنترل ورودی با یک الگوی تعریف شده توسط یک عبارت ، استفاده می گردد . کنترل فوق ، امکان بررسی لازم در خصوص دنباله کاراکترهای مورد انتظار نظیر آدرس های پست الکترونیکی ، شماره های تلفن و یا کد پستی را فراهم می نماید . با استفاده از خصلت ValidationExpression ، عبارت مورد نظر بمنظور بررسی صحت داده کنترل ورودی، تعریف و مشخص می گردد . گرامر عبارت بررسی صحت داده بر روی سرویس گیرنده و سرویس دهنده با یکدیگر متفاوت است . در سرویس گیرنده ، بمنظور مشخص نمودن عبارت مورد نظر از گرامر Jscript و بر روی سرویس دهنده از گرامر Regex استفاده می گردد . با توجه به اینکه، گرامرعبارات Jscript زیر مجموعه ای از گرامر Regex می باشد ، توصیه می گردد از گرامر Jscript برای تعریف عبارت مورد نظر خود استفاده تا شاهد نتایج یکسانی بر روی سرویس گیرنده و سرویس دهنده باشیم .

* کنترل CompareValidator . کنترل فوق، مقدار ورودی در یک کنترل ورودی را با مقدار دیگر مقایسه می نماید . از خصلت ControlToValidate برای مشخص نمودن کنترل ورودی اول و از خصلت ControlToCompare برای مشخص نمودن کنترل ورودی دوم که می بایست با یکدیگر مقایسه گردند ، استفاده می شود .درصورتیکه ماحصل مقایسه یکسان باشد، از مقدار مشخص شده توسط خصلت ControlToCompare ، استفاده می گردد .

استفاده از کنترل های اعتبار داده ASP.NET توسط ویژوال استودیو دات نت

بررسی صحت داده ورودی توسط کاربر در بخش رابط کاربر هر نرم افزار ، از جمله عملیاتی است که اولا" وقت زیادی را بخود اختصاص و ثانیا" در برخی موارد چالش های خاص خود را بدنبال خواهد داشت . فریمورک دات نت در این رابطه مجموعه ای از کنترل های بررسی صحت داده را ارائه که با استفاده از آنان می توان اقدام به بررسی داده در کنترل های ورودی موجود بر روی یک فرم وب و ارائه پیام خطاء مورد نظر در صورت عدم رعایت ضوابط و شرایط موجود توسط کاربر بمنظور ورود اطلاعات نمود.
در ادامه به بررسی مثالی خواهیم پرداخت که با ارائه یک فرم از کاربر درخواست تکمیل و ارسال اطلاعات خواهد شد. در این رابطه از کاربر درخواست نام ، آدرس پست الکترونیکی و رمز عبور می گردد . پس از ارسال اطلاعات توسط کاربر ، کنترل های بررسی صحت و اعتبار داده موجود بر روی فرم ، عملیات تائید داده ورودی توسط کاربر را انجام و درصورت عدم رعایت ضوابط تعریف شده در ارتباط با هر یک از کنترل های ورودی ، پیام خطائی ،بصورت مختصر و در قسمت انتهائی صفحه نمایش داده خواهد شد.

ایجاد یک برنامه وب ASP.NET با استفاده از VB.NET

بمنظور ایجاد یک برنامه وب ASP.NET با استفاده از VB.NET مراحل زیر را دنبال می نمائیم :

* اجرای برنامه ویژوال استودیو دات نت
* از طریق منوی File ، گزینه New و در ادامه Project را انتخاب نمائید .
* در جعبه محاوره ای New Project ، در بخش Project Types گزینه Visual Bacic Projects را انتخاب (کلیک) و در ادامه ASP.NET Web Application را انتخاب نمائید .

* در فیلد Location ، بجای نام پیش فرض #WebApplication ، نام ValidationData را در نظر می گیریم . در صورتیکه از یک سرویس دهنده محلی استفاده می گردد ، می توان نام سرویس دهنده را http://localhost در نظر گرفت . بدین ترتیب در فیلد Location ، آدرس مربوطه بصورت زیر نشان داده خواهد شد : http://localhost/ValidationData

ایجاد یک فرم وب نمونه

بمنظور ایجاد یک فرم وب در پروژه ایجاد شده در مرحله قبل ، مراحل زیر را دنبال می نمائیم :

* اضافه نمودن یک فرم جدید با نام ValidUser.aspx به برنامه وب ASP.NET در ویژوال استودیو دات نت . بمنظور انجام خواسته فوق ، دو مرحله زیر را دنبال می نمائیم :
مرحله یک : در Solution Explorer ، بر روی گره Project کلیک سمت راست نموده و گزینه Add و در ادامه گزینه Add Web Form انتخاب گردد.
مرحله دو : درفیلد Name ، نام ValidUser.aspx را درج و در ادامه گزینه Open انتخاب گردد .

* در ادامه، کدهای زیر را پس از استقرار در پنجره نمایش HTML مربوط به ValidUser.aspx ، بین تگ های شروع و پایان مستقر می نمائیم .

تگ های HTML درج شده در ValidUser.aspx

```
<table>
<tr width="100">
<td>نام</td>
<td><input id="txtUserName" type="text" size="20" maxlength="15" runat="server"
NAME="txtUserName">*
</td>
</tr>
<tr width="100">
<td>
آدرس پست الکترونیکی
<td><input id="txtEmail" type="text" size="35" maxlength="30" runat="server"
NAME="txtEmail">
(<A>Ino@Srco.ir</A>)
</td>
</tr>
<tr width="100">
<td>رمز عبور</td>
<td><input id="txtPassword" type="password" size="15" maxlength="10"
runat="server" NAME="txtPassword">*
</td>
</tr>
<tr width="100">
<td>تایپ مجدد رمز عبور</td>
<td><input id="txtConfirmPassword" type="password" size="15" maxlength="10"
runat="server" NAME="txtConfirmPassword">*
</td>
</tr>
</table>
```

* کنترل های RequiredFieldValidator ، بررسی لازم در خصوص داده ورودی برای هر یک از فیلدهای موردنظر در ارتباط با کنترل های مشخص شده را انجام خواهد داد. در این رابطه کنترل های RequiredFieldValidator را در ارتباط با فیلدهای UserName و Password به فرم اضافه می نمائیم . در ValidUser.aspx ، و پنجره HTML مستقر و کد زیر را بعد از تگ </Table> اضافه می نمائیم .

```
<asp:RequiredFieldValidator id=valUserNameRequired
ControlToValidate=txtUserName ErrorMessage="نام کاربر می بایست وارد شود"
EnableClientScript=true Display=None Runat=server/>
<asp:RequiredFieldValidator id=valPasswordRequired
ControlToValidate=txtPassword ErrorMessage="رمز عبور می بایست وارد شود"
EnableClientScript=true Display=None Runat=server/>
<asp:RequiredFieldValidator id=valConfirmPasswordRequired
ControlToValidate=txtConfirmPassword ErrorMessage=" تائید مجدد رمز عبور می بایست
وارد شود"
EnableClientScript=true Display=None Runat=server/>
```

* فیلدهای رمز عبور تاکید مضاعفی است که کاربر دومرتبه و بدرستی رمز عبور خود را وارد نماید. کنترل CompareValidator محتویات دو فیلد را با یکدیگر مقایسه و در صورت عدم یکسان بودن آنان ، پیام خطائی نمایش داده خواهد شد . در این رابطه از یک کنترل CompareValidator بمنظور بررسی صحت (یکسان بودن) فیلدهای رمز عبور استفاده شده است . بدین منظور پس از استقرار در پنجره HTML مربوط به فرم وب ValidUser.aspx ، کد زیر را پس از کنترل های اضافه شده در مرحله قبل ، به فرم مورد نظر اضافه می نمائیم .

```
<asp:CompareValidator id=valComparePassword
ControlToValidate=txtConfirmPassword ErrorMessage="فیلدهای رمز عبور می بایست یکسان باشند" ControlToCompare=txtPassword Display=None
EnableClientScript=true Runat=server/>
```

* در برخی موارد لازم است از نوع خاصی بررسی صحت و اعتبار داده استفاده گردد . فیلد آدرس پست الکترونیکی در مثال فوق ، نمونه ای در این زمینه است . در این رابطه از کنترل RegularExpressionValidator استفاده تا این اطمینان حاصل گردد که کاربران فرمت اولیه برای درج یک آدرس پست الکترونیکی را رعایت نموده اند . محتویات فیلد فوق مجددا" بر اساس یک عبارت (تعریف یک الگو) بررسی شده و در صورتیکه محتویات مورد نظر با عبارت تعریف شده مطابقت نداشته باشد ، یک پیام خطاء نمایش داده خواهد شد. در این رابطه لازم است که یک کنترل RegularExpressionValidator بر روی فرم اضافه تا فرمت داده آدرس پست الکترونیکی ورودی توسط کاربر را بررسی و از صحت و رعایت فرمت مورد نظر اطمینان حاصل گردد . بدین منظور پس از استقرار در پنجره HTML مربوط به فرم وب ValidUser.aspx ، کد زیر را پس از کنترل های اضافه شده در مرحله قبل ، به فرم مورد نظر اضافه می نمائیم .

```
<asp:RegularExpressionValidator ID=valEmailAddress
ControlToValidate=txtEmail ValidationExpression=".*@.*\..*" ErrorMessage="آدرس پست الکترونیکی درست نمی باشد"
Display=None EnableClientScript=true Runat=server/>
```

* در ادامه ، بر روی فرم ValidUser.aspx ، یک دکمه "ارسال" را اضافه می نمائیم .بدین ترتیب به کاربر اجازه داده خواهد شد تا صفحه مورد نظر را برای سرویس دهنده ارسال و عملیات بررسی صحت و اعتبار داده درج شده در هر یک از فیلدهای موجود بر روی فرم ، انجام شود. بدین منظور پس از استقرار در پنجره HTML مربوط به فرم وب ValidUser.aspx ، کد زیر را پس از کنترل های اضافه شده در مرحله قبل ، به فرم مورد نظر اضافه می نمائیم .

```
<br>
<input type=submit id=cmdSumbit value=submit runat=server/>
```

* در نهایت ، از یک کنترل ValidationSummary بمنظور نمایش تمامی خطاهای بوجود آمده در یک ناحیه خاص بر روی فرم استفاده می شود. بدین منظور پس از استقرار در پنجره HTML مربوط به فرم وب ValidUser.aspx ، کد زیر را پس از دکمه "ارسال" اضافه شده در مرحله قبل ، به فرم مورد نظر اضافه می نمائیم .

```
<br><br>
<asp:ValidationSummary id=ValSummary HeaderText="The following
errors were found:" ShowSummary=True DisplayMode=List Runat=server/>
```

* پس از اتمام مراحل فوق ، با انتخاب گزینه Save ، فرم وب ایجاد شده ذخیره و در ادامه می توان با استفاده از منوی Debug و گزینه Start ، امکان ایجاد و اجرای برنامه وب را فراهم نمود.

اجرای برنامه

در صورتیکه کاربر دکمه "ارسال" را بدون درج هیچگونه اطلاعاتی در فیلدهای مربوطه ، فعال نماید ، سه پیام خطاء بصورت زیر نمایش داده خواهد شد .

در صورتیکه کاربر، دو رمز عبور را وارد که با یکدیگر یکسان نمی باشند ، پیام خطاء زیر ارائه خواهد شد .

در صورتیکه کاربر یک آدرس پست الکترونیکی را وارد که دارای فرمت مناسب نباشد ، ، پیام خطاء زیر نمایش داده خواهد شد .

دستیابی به بانک اطلاعاتی Access با استفاده از VB.NET

در این مقاله قصد داریم به نحوه بازیابی و نمایش اطلاعات موجود در یک بانک اطلاعاتی Access اشاره نمائیم . هدف از مقاله فوق ، پرداختن به تمامی رویکردهای موجود در این زمینه نبوده و صرفا" به معرفی یکی از گزینه های موجود در این زمینه اشاره خواهد شد. در این راستا از تکنولوژی های ASP.NET، ADO.NET و VB.NET استفاده خواهد شد . از کلاس های OleDbConnection، OleDbCommand و OleDbDataReader مربوط به ADO.NET بمنظور انجام عملیات لازم در ارتباط با بانک اطلاعاتی ، از ASP.NET بمنظور ایجاد فرم وب و ارائه داده با استفاده از کنترل سرویس دهنده Table و از زبان VB.NET بمنظور نوشتن دستورالعمل های مورد نظر استفاده می گردد .
در ابتدا لازم است با سه کلاس ADO.NET که در ادامه از آنان استفاده خواهد شد ، بیشتر آشنا شویم :

* کلاس OleDbConnection . شی فوق ، یک اتصال منحصر بفرد با یک منبع داده را ایجاد می نماید. در رابطه با یک بانک اطلاعاتی سرویس گیرنده / سرویس دهنده ، این امر معادل یک اتصال شبکه به سرویس دهنده است . با توجه به قابلیت های حمایت شده توسط native OLE DB Provider ، برخی از متدها و یا خصلت ها مربوط به شی OleDbConnection ممکن است در دسترس و قابل استفاده نباشد . زمانیکه نمونه ای از OleDbConnection ایجاد می گردد ، تمامی خصلت های مربوطه ، مقدار اولیه خود را دارا خواهند بود . پس از اتمام عملیات موردنظر در ارتباط با بانک اطلاعاتی ، می بایست با فراخوانی Close و یا Dispose اقدام به غیر فعال نمودن اتصال ایجاد شده با بانک اطلاعاتی مربوطه نمود.

* کلاس OleDbCommand . یک عبارت SQL و یا Stored procedure را بمنظور اجراء در رابطه با یک منبع داده ارائه می نماید. کلاس فوق از متدهای زیر بمنظور اجرای دستورات در رابطه با یک منبع داده استفاده می نماید.
ExecuteReader . متد فوق ، دستوراتی را اجراء می نماید که خروجی آنان شامل سطرهائی خواهد بود.
ExecuteNonQuery . باعث اجرای دستوراتی نظیر INSERT,DELETE,UPDATE SQL و SET خواهد شد .
ExecuteScalar . بازیابی صرفا" یک مقدار از یک بانک اطلاعاتی

* کلاس OleDbDataReader . متد فوق ، امکان خواندن سطرهائی از داده موجود در یک منبع داده را فراهم می نماید(فقط بسمت جلو) . بمنظور ایجاد یک OleDbDataReader ، می بایست متد ExecuteReader مربوط به شی OleDbCommand فراخوانده شود. مادامیکه

OleDbDataReader در حال استفاده است (اتصال مرتبط OleDbConnection) ، عملیات دیگری را در ارتباط با OleDbConnection نمی توان انجام داد .

امکانات مورد نیاز

برای دنبال نمودن این مقاله و اجرای نمونه مثالی که در ادامه بررسی می گردد ، به امکانات زیر نیاز خواهد بود :

* نصب یکی از نسخه های ویندوز ۲۰۰۰ و یا نسخه ویندوز ۲۰۰۳
* نصب IIS
* نصب فریمورک دات نت نسخه ۱,۰ ,یا نسخه ۱,۱
* یک بانک اطلاعاتی نمونه اکسس نظیر Northwind

ایجاد یک برنامه وب ASP.NET با استفاده از VB.NET

بمنظور ایجاد یک برنامه وب ASP.NET با استفاده از VB.NET مراحل زیر را دنبال می نمائیم :

* اجرای برنامه ویژوال استودیو دات نت
* از طریق منوی File ، گزینه New و در ادامه Project را انتخاب نمائید .
* در جعبه محاوره ای New Project ، در بخش Project Types گزینه Visual Bacic Projects را انتخاب (کلیک) و در ادامه ASP.NET Web Application را انتخاب نمائید .
* در فیلد Location ، بجای نام پیش فرض #WebApplication ، نام TestAccessDB را انتخاب نمائید . در صورتیکه از یک سرویس دهنده محلی استفاده میگردد ، می توان نام سرویس دهنده را http://localhost در نظر گرفت . بدین ترتیب در فیلد Location ، آدرس مربوطه بصورت زیر نشان داده خواهد شد : http://localhost/TestAccessDB

ایجاد یک فرم وب نمونه

در نمونه کد نوشته شده از کنترل سرویس دهنده Table مربوط به ASP.NET استفاده شده که بصورت پویا یک نمایش ساده از داده بازیابی شده را نشان خواهد داد. ASP.NET ، مجموعه متنوعی از کنترل های انعطاف پذیر را ارائه که می توان از آنان با توجه به رویکردهای متفاوت در رابطه با نمایش داده استفاده نمود. بمنظور ایجاد یک فرم وب در پروژه ایجاد شده در مرحله قبل ، مراحل زیر را دنبال می نمائیم :

* اضافه نمودن یک فرم جدید با نام DataSample.aspx به برنامه وب ASP.NET در ویژوال استودیو دات نت . بمنظور انجام خواسته فوق ، دو مرحله زیر را دنبال می نمائیم :
مرحله یک : در Solution Explorer ، بر روی گره Project کلیک سمت راست نموده و گزینه Add و در ادامه گزینه Add Web Form انتخاب گردد.
مرحله دو : درفیلد Name ، نام DataSample.aspx را درج و در ادامه گزینه Open انتخاب گردد .

* از طریق Toolbox مربوط به Web Forms Tab ، یک Table ASP.NET Server Control را انتخاب (Drag) و بر روی صفحه aspx . مستقر نمائید (در حالت Desgin view) .

* در Properties نام ID را به DisplayTable تغییر دهید .

* در Solution Explorer ، بر روی صفحه aspx . ، کلیک سمت راست نموده و گزینه View Code را انتخاب نمائید .

* مرجع namespace زیر را در بالاترین قسمت فایل کلاس code-behind وارد نمائید .

```
Imports System.Data.OleDb
```

* کد زیر را در ارتباط با رویداد Page_load در نظر می گیریم :

Page_Load Event handler

```
Private Sub Page_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load

'استفاده از یک متغیر رشته ای بمنظور ذخیره سازی Connectionstring

  Dim connectString As String = "Provider=Microsoft.Jet.OLEDB.4.0;" &_
  "Data Source=C:\MyDB\NWIND.MDB"

' ایجاد یک شی OleDbConnection و ارسال مقدار متغیر ConnectionString

  Dim cn As OleDbConnection = New OleDbConnection(connectString)

'فعال نمودن Connection

  cn.Open()

'استفاده از یک متغیر بمنظور ذخیره عبارت SQL

  Dim selectString As String = "SELECT CustomerID, ContactName, Phone FROM
Customers"

'ایجاد یک شی OledbCommand
' ارسال می گردد OleDbConnection و شی SQL در این خط متغیر عبارت

  Dim cmd As OleDbCommand = New OleDbCommand(selectString, cn)

' ایجاد یک OleDbDataReader و ارسال CommandText به Connection
'از نوع " فقط بسمت جلو " خواهد بود OleDbDataReader

  Dim reader As OleDbDataReader = cmd.ExecuteReader()

' تنظیم عرض جدول

  DisplayTable.Width = Unit.Percentage(90.0)

'ایجاد یک سطر جدید برای اضافه نمودن عنوان جدول

  Dim tableHeading As TableRow = New TableRow()
```

```
' ایجاد و اضافه نمودن سلول های شامل ستون Customer ID

  Dim customerIDHeading As TableHeaderCell = New TableHeaderCell()
  customerIDHeading.Text = "Customer ID"
  customerIDHeading.HorizontalAlign = HorizontalAlign.Left
  tableHeading.Cells.Add(customerIDHeading)

' ایجاد و اضافه نمودن سلول های شامل ستون Contact Name

   Dim contactNameHeading As TableHeaderCell = New TableHeaderCell()
   contactNameHeading.Text = "Contact Name"
   contactNameHeading.HorizontalAlign = HorizontalAlign.Left
   tableHeading.Cells.Add(contactNameHeading)

' ایجاد و اضافه نمودن سلول های شامل ستون Phone

    Dim phoneHeading As TableHeaderCell = New TableHeaderCell()
    phoneHeading.Text = "Phone"
    phoneHeading.HorizontalAlign = HorizontalAlign.Left
    tableHeading.Cells.Add(phoneHeading)

    DisplayTable.Rows.Add(tableHeading)

' تکرار در بین داده انتخابی نتایج و افزودن داده برای هر یک از ستون های مورد نظر در جدول

    While(reader.Read())
      Dim detailsRow As TableRow = New TableRow()
      Dim customerIDCell As TableCell = New TableCell()
      customerIDCell.Text = reader("CustomerID").ToString()
      detailsRow.Cells.Add(customerIDCell)
      Dim contactNameCell As TableCell = New TableCell()
      contactNameCell.Text = reader("ContactName").ToString()
      detailsRow.Cells.Add(contactNameCell)
      Dim phoneCell As TableCell = New TableCell()
      phoneCell.Text = reader("Phone").ToString()
      detailsRow.Cells.Add(phoneCell)
      DisplayTable.Rows.Add(detailsRow)
    End While

' بستن Connection

    reader.Close()
   cn.Close()
End Sub
```

* مقدار متغیر ConnectString در ابتدای کد نوشته شده فوق را تغییر و آن را به محلی که بانک اطلاعاتی موجود است ، اشاره دهید .

* از طریق منوی File ، گزینه Save All را انتخاب تا فرم وب و سایر فایل های مرتبط با پروژه ، ذخیره گردد .

* از طریق منوی Build ، گزینه Build Solution را بمنظور ایجاد پروژه ، فعال نمائید.

* در Solution Explorer ، بر روی DataSample.aspx کلیک سمت راست و در ادامه گزینه View in Browser را انتخاب نمائید . در ادامه صفحه در مرورگر نمایش و شامل داده موجود در بانک اطلاعاتی مربوطه است .

اشکالات و خطاهای احتمالی

درزمان اجراء ، ممکن است با خطائی مانند زیر مواجه شویم :

The Microsoft Jet database engine cannot open the file 'C:\MyDB\NWIND.MDB'.
It is already opened exclusively by another user, or you need permission to view its data.

* خطای فوق ، اغلب بدلیل عدم داشتن مجوز لازم بمنظور دستیابی به فایل (فایل بانک اطلاعاتی با انشعاب mdb .) می باشد . بصورت پیش فرض ، ASP.NET تحت ASPNET account در فریمورک دات نت نسخه یک و یا NetworkService در فریمورک دات نت نسخه ۱،۱ اجراء می گردد.در این رابطه لازم است تغییرات لازم در رابطه با مجوز دستیابی به فایل mdb . و فولدری که شامل فایل است ، اعمال گردد .

* از نصب عناصر مورد نیاز Microsoft Jet database بر روی سیستم ، مطمئن گردید. در صورتیکه MDAC 2.6 نصب شده باشد (Microsoft Data Access Components 2.6) ، عناصر Jet را شامل نمی گردد . در چنین مواردی می بایست آخرین نسخه Jet 4.0 Service Pack ، بر روی سیستم نصب گردد.

معرفی اولیه VB.NET

یک برنامه کامپیوتری ، از مجموعه ای دستورالعمل که نوع و نحوه انجام یک فعالیت را برای کامپیوتر مشخص می نمایند ، تشکیل می گردد. دستورالعمل های نوشته شده بعنوان نمونه ممکن است به کامپیوتر اعلام نمایند که تعدادی از اعداد را با یکدیگر جمع و یا دو عدد را بایکدیگر مقایسه و بر اساس نتیجه بدست آمده ، اتخاذ تصمیم نماید. دستورالعمل های نوشته شده ، صرفا" برای کامپیوتر قابل فهم و اجراء خواهند بود.کامپیوتر دستورالعمل های نوشته شده را اجراء و ماحصل آن رسیدن به اهدافی خواهد بود که بر اساس آن برنامه طراحی و پیاده سازی شده است . دستورالعمل ها ، می بایست با استفاده از یکی از زبانها ی برنامه نویسی نوشته شده (کد ینگ) و در ادامه در اختیار کامپیوتر قرار داده شوند. زبانهای برنامه نویسی متعددی نظیر : فرترن ، بیسیک ، کوبال ، پاسکال، C ، جاوا ، ویژوال بیسیک و ... وجود دارد.

برنامه نویسی کامپیوتر ، مشابه آموزش گره زدن کفش به کودکان است . برای نیل به هدف فوق ، می بایست تمامی مراحل لازم بصورت شفاف به کودکان آموزش داده شود . کودکان با

دنبال نمودن دستورالعمل های ارائه شده ، قادر به گره زدن کفش خود خواهند بود (روش انجام این کار برای آنان مشخص شده و بر اساس آن ، امکان نیل به هدف مورد نظر توسط کودکان فراهم می گردد) . VB.NET ، زبانی است که می توا ن نحوه نیل به یک خواسته را بکمک آن بصورت شفاف (نظیر آموزش گره زدن کفش به کودکان) مشخص و کامپیوتر با دنبال نمودن مراحل مشخص شده ، خواسته مورد نظر را محقق نماید. با استفاده از VB.NET ، می توان محصولاتی راایجاد که زمینه استفاده از آنان در محیط ویندوز و اینترنت ، وجود خواهد داشت . فراموش نکنیم در زمان فراگیری یک تکنولوژی در ابتدا می بایست شیوه راه رفتن را بیاموزیم و در ادامه اقدام به دویدن نمود .

VB.NET یکی از زبان های حمایت شده در دات نت می باشد . با استفاده از زبان فوق علاوه بر اینکه می توان برنامه های مبتنی بر ویندوز را پیاده سازی نمود ، امکان استفاده از آن بعنوان زبان مورد نظر در زمان ایجاد برنامه های مبتنی بر وب که از تکنولوژی ASP.NET استفاده می نمایند ، نیز وجود خواهد داشت . با توجه به اهمیت و جایگاه خاص این زبان در دات نت ، مجموعه مقالاتی در رابطه با آموزش اصولی این زبان آماده شده که بتدریج بر روی سایت قرار خواهند گرفت . در اولین مقاله از این مجموعه به معرفی اولیه VB.NET خواهیم پرداخت . در ابتدا لازم است با ویژگی های منحصر بفرد برنامه های مبتنی بر ویندوز در قیاس با برنامه های مبتنی بر DOS ، آشنا شده و پس از مروری مختصر به روند شکل گیری نسخه های متعدد ویژوال بیسیک ، با نحوه نصب آن نیز آشنا شویم .

برنامه نویسی مبتنی بر DOS در مقابل ویندوز

برنامه نویسی مبتنی بر ویندوز دارای تفاوت های عمده ای نسبت به برنامه نویسی سنتی در محیط DOS است.برنامه های DOS ، مسیری دقیق و مشخص را ازابتدا تا پایان دنبال می نمایند . رویکرد فوق ، باعث بروز محدودیت هائی در رابطه با عملکرد برنامه ها از یکطرف و تحمیل محدودیت هائی به کاربران در طی نمودن مسیر مشخص شده ، می گردد. از زاویه ای خاص می توان عملکرد یک برنامه مبتنی بر DOS را مشابه قدم زدن در یک راهرو (سالن)، در نظر گرفت . بمنظور رسیدن به نقطه انتهائی سالن ، می بایست طول سالن طی تا به انتهای آن رسید . در این راستا از موانع متعدد موجود در مسیر ، می بایست عبور تا سرانجام به مقصد مورد نظر رسید . در زمان پیمودن مسیر ، صرفا" امکان بازنمودن درب های خاصی ، وجود خواهد داشت . ویندوز ، دنیای جدیدی از برنامه نویسی مبتنی بر "رویداد" را ایجاد نموده است . کلیک نمودن موس ، تغییر اندازه پنجره ، تغییر محتویات یک Textbox ، نمونه هائی از یک "رویداد" می باشند. کدهای نوشته شده ، نحوه برخورد با یک رویداد را مشخص می نماید. برای رسیدن به انتهای یک سالن کافی است بر روی "انتهای سالن " ، کلیک نمود و دیگر ضرورتی به پیمودن تمامی مسیر تا رسیدن به انتهای سالن نخواهد بود . درصورتیکه به انتهای سالن رسیده باشیم و متوجه گردیم که این مکان ، محلی نیست که انتظار آن را داشته ایم ، بسادگی می توان مقصد جدیدی را برای خود انتخاب ، بدون اینکه ضرورتی به برگشت در نقطه آغازین مسیر وجود داشته باشد. برنامه نوشته شده عکس العمل های لازم در ارتباط با حرکت شما را بهمراه عملیات مربوطه بمنظور تکمیل فعالیت ها ی مورد نظر انجام خواهد داد . با استفاده از VB.NET ، می توان کدهای لازم بمنظور ارائه عکس العمل لازم در زمان تحقق یک رویداد را نوشت .در این راستا ، برنامه نویسان می توانند کدهای لازم در رابطه با رویدادهائی که امکان تحقق آنها وجود دارد را نوشته تا در زمان بروز رویداد مورد نظر ، عکس العمل لازم از طرف برنامه صورت پذیرد.در این زمینه می توان از نوشتن کدهای دیگر بمنظور برخورد با رویدادهای غیرضروری ، صرفنظر کرد. مثلا" ویندوز قادر به تشخیص رویداد"کلیک " از "کلیک مضاعف " است . این بدان معنی است که اگر می خواهید برنامه مورد نظر شما ،عکس العمل لازم در ارتباط با رویداد "کلیک" را داشته باشد ، می بایست صرفا" کد مربوط به رویداد " کلیک"، نوشته گردد و الزامی به نوشتن کدهای لازم بمنظور برخورد با رویداد "کلیک مضاعف" ، وجود نخواهد داشت . در دنیای برنامه نویسی DOS ، کاربر عکس العمل لازم را نسبت به برنامه انجام

می دهد در صورتیکه در ویندوز ، برنامه ها عکس العمل لازم را با توجه به رفتار کاربران ، انجام خواهند داد .
یکی دیگر از مزایای مهم برنامه های ویندوز ، عدم وابستگی برنامه ها به یک سخت افزار خاص است . ویندوز تمهیدات لازم در خصوص ارتباط با سخت افزار را پیش بینی و برنامه نویسان نیاز به آگاهی از نحوه عملکرد یک دستگاه سخت افزاری خاص بمنظور استفاده از آن ، نخواهند داشت . مثلا" برنامه نویسان ضرورتی به آگاهی از نحوه عملکرد هر نوع چاپگر لیزری، بمنظور ایجاد خروجی مورد نظر خود در برنامه ها ، نخواهند داشت. ویندوز، امکانات لازم در این خصوص را از طریق ارائه روتین های عمومی که با درایورهای مورد نظر مرتبط می گردند ، فراهم می نماید. شاید همین موضوع دلیل موفقیت ویندوز باشد .
روتین های عمومی اصطلاحا" Windows (API) Application Programming Interface نامیده می شوند .

تاریخچه ویژوال بیسیک

قبل از معرفی ویژوال بیسیک در سال ۱۹۹۱ ، پیاده کنندگان نرم افزار مجبور به تسلط و مهارت در زمینه استفاده از C++ بهمراه موارد پیچیده ای در این خصوص بودند . بدین ترتیب ، صرفا" افراد خاص آموزش دیده، قادر به خلق نرم افزارهای قدرتمند بمنظور اجراء 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 ، زمانیکه بخش رابط کاربر ، ترسیم می گردد ، برنامه نویس می تواند کدهای لازم بمنظور انجام عکس العمل مناسب در رابطه با رویداد ها را به آن اضافه نماید . زمانیکه ماکروسافت نسخه شماره سه ویژوال بیسیک را ارائه نمود ، مجددا" دنیای برنامه نویسی با تغییر مهمی مواجه گردید. در این راستا امکانات مناسبی برای نوشتن برنامه های مبتنی بر بانک های اطلاعاتی ، در اختیار برنامه نویسان قرار گرفت. ماکروسافت بدین منظور محصول جدیدی با نام Data Access Objects(DAO) را ارائه نمود . برنامه نویسان با استفاده از DAO ، امکان انجام عملیات متفاوت در رابطه با داده ها را ، بدست 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 اجراء در محیط ویندوز ۹۵ ، طراحی و پیاده سازی نمایند . در این زمینه ، برنامه نویسان قادر به نوشتن کدهائی گردیدند که امکان استفاده از آنان توسط سایر پیاده کنندگان نرم افزار که از زبانی دیگر استفاده می کردند، فراهم گردید. نسخه شماره شش ویژوال بیسیک ، روش جدیدی بمنظور دستیابی به بانک های اطلاعاتی را ارائه نمود: ActiveX Data Objects(ADO) . یکی از اهداف اولیه طراحی ADO ، امکان دستیابی به بانک های اطلاعاتی برای پیاده 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 شود ، بسیاری از محدودیت های مرتبط با ویژوال بیسیک برطرف گردید . در گذشته ویژوال بیسیک با انتقادات فراوان مواجه (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 ، بعنوان یک ابزار پیاده سازی بسیار قدرتمند مطرح و گزینه ای مناسب برای برنامه نویسان در تمامی سطوح باشد .

نصب VB.NET

برای نصب VB.NET ، از دو رویکرد متفاوت می توان استفاده کرد :

* نصب بهمراه ویژوال استودیو دات نت

* نصب نسخه استاندارد

هر یک از گزینه های فوق ، امکان ایجاد برنامه های مبتنی بر ویندوز را فراهم می نمایند . مراحلی که در ادامه ذکر می گردد ، نحوه نصب ویژوال استودیو را تشریح می نماید .

*** مرحله اول :** برنامه Setup.exe را از روی CD مربوطه فعال نمائید.

*** مرحله دوم :** جعبه محاوره ای ، مراحل و اولویت های عملیات نصب را نشان خواهد داد. بمنظور صحت عملکرد VB.NET ، چندین Component نصب و یا بهنگام خواهند شد . اولین مرحله نصب، بهنگام سازی عناصر (Components) است . بر روی گزینه Windows Component Update ، کلیک نمائید.

*** مرحله سوم :** برنامه نصب در ادامه سیستم را بررسی تا نوع عناصری را که می بایست بهنگام گردند، مشخص گردد. دامنه فرآیند بهنگام سازی به وضعیت ماشینی که بر روی آن ویژوال استودیو دات نت نصب می گردد، بستگی خواهد داشت .

*** مرحله چهارم :** با توجه به اینکه ممکن است در زمان بهنگام سازی لازم باشد چندین مرتبه سیستم راه اندازی گردد ، از شما درخواست نام و رمز عبور شده تا ضرورتی به نشستن و نگاه کردن به کامپیوتر و واکنش لازم (درج نام و رمز عبور به سیستم) پس از هر مرتبه راه اندازی سیستم نباشد . بدین ترتیب در زمان راه اندازی سیستم ، عملیات مربوطه بصورت اتوماتیک و بدون نیاز به تایپ نام و رمز عبور ، انجام خواهد شد . عملیات فوق ، اختیاری است و در صورتیکه گزینه فوق انتخاب نگردد ، با هر مرتبه راه اندازی سیستم، پیام مناسب ارائه و می بایست واکنش لازم (تایپ نام و رمز عبور) را انجام داد .

*** مرحله پنجم :** در این مرحله با فشردن دکمه ! Install Now ، بهنگام سازی عناصر (Components) آغاز می گردد . با اتمام هر یک از آیتم ها یک Check mark بمنزله اتمام مرحله مربوطه نشان داده می شود . در مقابل عنصر جاری برای بهنگام سازی نیز یک فلش قرمز رنگ نسان داده می شود.

*** مرحله ششم :** پس از بهنگام سازی عناصر ، مجددا" به صفحه اصلی Setup مراجعت و امکان نصب ویژوال استودیو دات نت فراهم می گردد.(کلیک نمودن برروی گزینه Visiual Studio.NET)
نکته : در صورتیکه قصد دارید که از طریق ماشین فوق ، یک برنامه تحت وب پیاده سازی نمائید ، لازم است IIS و FrontPage Extensions قبلا" نصب شده باشد(بصورت پیش فرض در زمان نصب ویندوز ۲۰۰۰ نصب خواهد شد) در صورتیکه ویژوال استودیو دات نت ، بر روی کامپیوتری نصب می گردد که دارای سیستم عامل ویندوز ۲۰۰۰ نسخه Professional است ، با یک پیام خطاء مواجه خواهیم شد(عدم وجود عناصر لازم) با فشردن دکمه Install Component ، عملیات نصب IIS و Frontpage Extensions انجام خواهد شد . در صورتیکه دکمه Continue ، انتخاب گردد ، در آینده نمی توانید برنامه های تحت وب را بصورت محلی بر روی کامپیوتر خود پیاده سازی نمائید .

*** مرحله هفتم :** نظیر اکثر برنامه های نصب ، لیستی از گزینه های موجود (شامل عناصر) برای نصب در اختیار شما قرار می گیرد . شما می توانید ، صرفا" آنچیزی را که بدان نیاز دارید ، نصب نمائید . مثلا" در صورتیکه ظرفیت درایو شما پایین و یا ضرورتی به استفاده از ویژوال C++ دات نت را ندارید ، می توان در این مرحله از نصب آن صرفنظر کرد. هر گزینه ای که در این مرحله انتخاب نمی گردد ، می توان در صورت ضرورت آن را در آینده نصب کرد. برای هر یک از امکاناتی که قرار است نصب گردند ، سه بخش اطلاعاتی متفاوت نمایش داده می شود :

بخش Feature Properties . فایل ها ی مورد نظر برای نصب و میزان فضای مورد نیاز را نشان می دهد .
بخش Feature description . هر Feature چیست و چه عملیاتی را انجام می دهد .
بخش Space Allocation ، وضعیت فضای ذخیره سازی هارد را با توجه به گزینه های انتخاب شده ، نشان خواهد داد .
نکته : زمانیکه ویژوال استودیو دات نت ، اجراء می گردد مجموعه ای از اطلاعات بین دیسک و حافظه مبادله می گردد . بنابراین لازم است به میزان کافی ظرفیت آزاد بر روی هارد دیسک وجود داشته باشد ، در این راستا نمی توان دقیقا" مشخص نمود که به چه میزان فضای آزاد نیاز خواهد بود ولی حداقل یکصد مگابایت توصیه می گردد .

* **مرحله هشتم :** ویژوال استودیو دات نت ، شامل مجموعه ای گسترده از فایل های مستندات (راهنما) است . در این مرحله می توان تنظیمات لازم در خصوص اجرای مستندات از طریق CD و یا دایرکتوری نصب شده بر روی هارد را انجام داد . در این زمینه می توان یک مسیر بر روی هارد را مشخص تا مستندات نصب و یا گزینه Run From Source را انتخاب تا بر اساس آن مستندات همچنان بر روی CD باقی بمانند .

* **مرحله نهم :** پس از انتخاب عناصر مورد نظر برای نصب ، با فشردن دکمه ! Install Now ، عملیات نصب آغاز می گردد . مدت زمان نصب ، بستگی به موارد انتخابی و نوع سیستم دارد . مثلا" نصب تمام ویژوال استودیو دات نت بهمراه تمامی مستندات بر روی یک ماشین با دارا بودن ۲۵۶ مگابایت حافظه اصلی، سرعت ۶۵۰ مگاهرتز و دوازده گیگابایت هارد دیسک ، حدود یک ساعت طول خواهد کشید .

* **مرحله دهم :** پس از اتمام مرحله قبل ، با انتخاب گزینه Service Release ، بررسی لازم در خصوص بهنگام سازی انجام می گیرد . این عملیات از طریق اینترنت انجام خواهد شد . در این زمینه به یک خط پرسرعت و مطمئن نیاز خواهد بود .