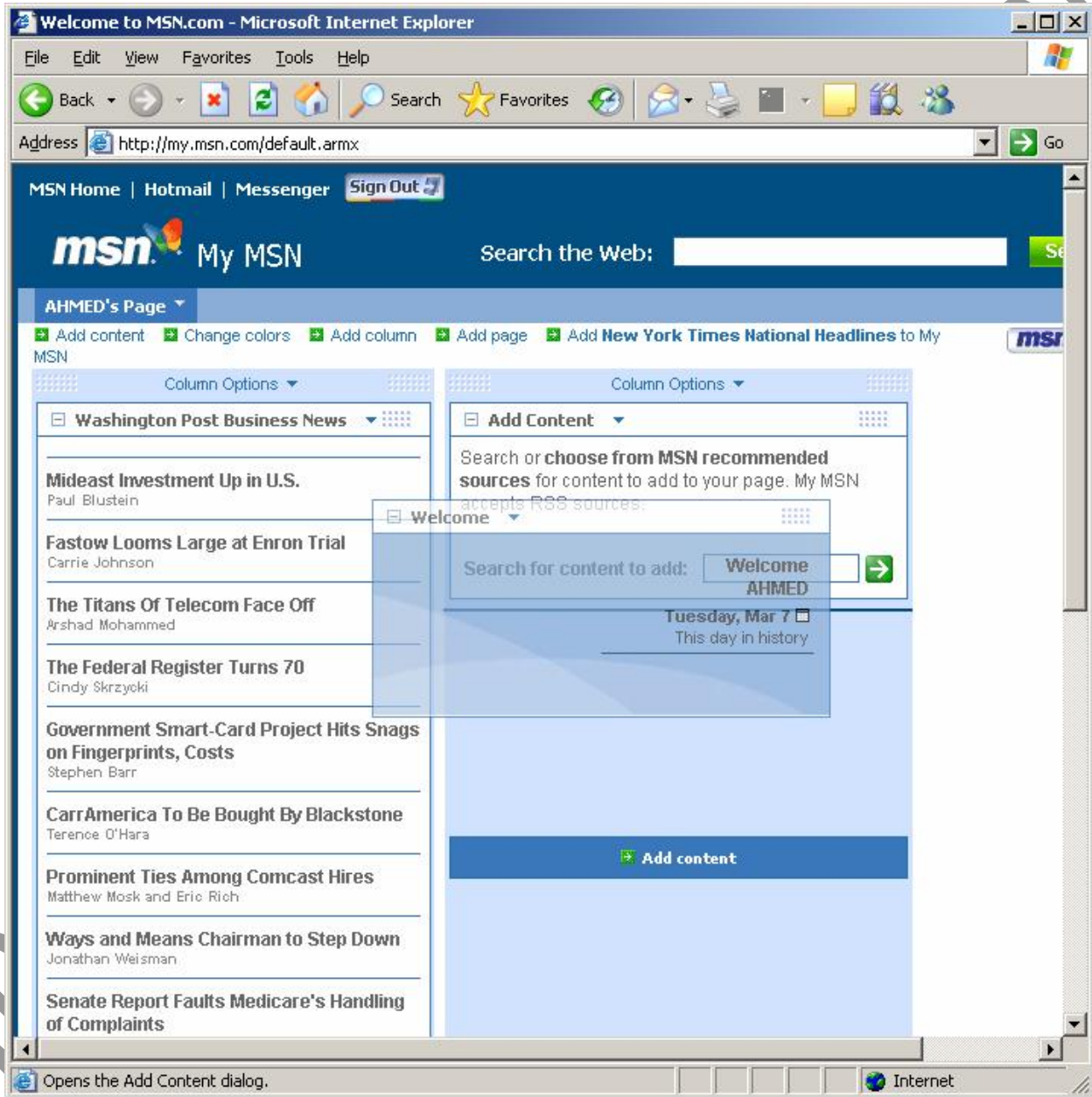


الفصل الثالث : أجزاء الويب و التفاعليه العاليه مع المستخدم.

## Web Parts, and rich user interface

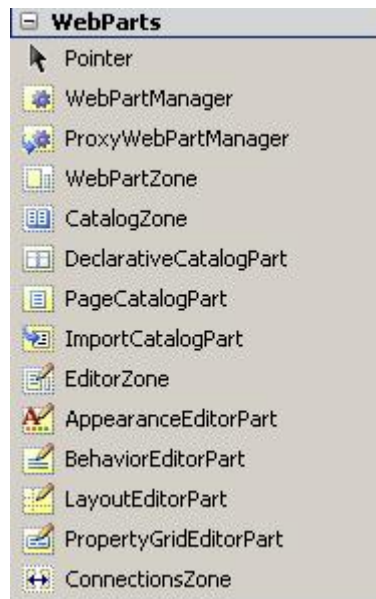
---

في هذا الفصل سوف نرى منطق جديد يسهل و يوفر على المطور الكثير من الوقت لعمل شئ مشابه، و لتوضيح فكرة أجزاء الويب، كل ما عليك هو الدخول على موقع <http://www.msn.com> و تقوم بعمل Sign in و بعد ذلك سوف تجد أنه من الممكن تحريك مكونات الصفحة و تعديلها كما تريد و سوف تلاحظ أنه يمكنك تغيير و تعديل كل شئ . و كما نلاحظ في الشكل التالي [1] أنه يمكنك نقل المكونات من مكان الى آخر بالإضافة إلى إمكانية تغيير الألوان .



الشكل 1

يوضح الشكل التالي [2] مجموعة الأدوات الجديدة الخاصة بهذا الموضوع، هذه الأدوات لها طريقة معينة للتعامل ليس مجرد سحب و أفلات للعناصر على الصفحة ، فهو ليس بالبسيط أوالمعقد يمكننا أن نقول "السهل الممتنع" ، لكن لا تقلق فالموضوع إذا تم تناوله بتسلسله الطبيعي سوف نصل إلى نتائج ممتازة جدا أكثر مما كنت تتوقع .



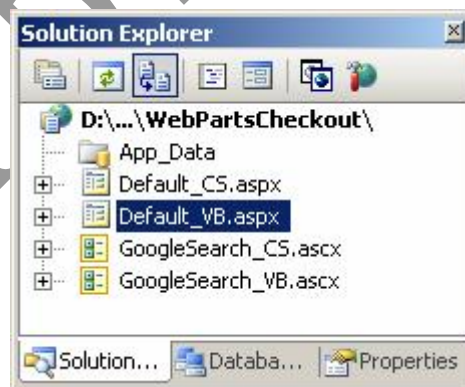
الشكل ٢

الآن لابد من معرفة بعض النقاط و الخطوات المهمة للتعامل مع هذه الأدوات و سوف يتم شرح الأدوات واحدة تلو الأخرى بكل متطلباتها ؛ وسيتم توضيح كل أداة من الأدوات بمثال توضيح لها .

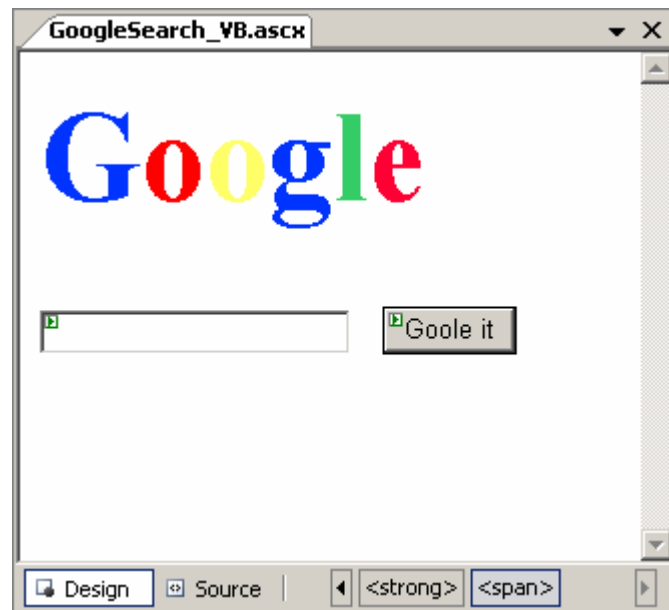
الآن سوف نبدأ بمثال صغير و سوف نستعمل فيه كلاً من الأدوات (WebPartManager) و (WebPartZone) لكن أولاً لا بد من توضيح شي بالنسبة لكل من الأدوات :-

- (WebPartManager) : تقوم هذه الأداة بإدارة التعامل مع أجزاء الويب الموجودة على الصفحة.
- (WebPartZone) : تقوم هذه الأداة بأحتواء الأدوات التي سيتم التحكم فيها.

وسيتم إيضاح التعريف السابق بمثال توضيحي لما يمكن عمله مع أجزاء الويب.



الشكل ٣ [ قم بإضافة الملفات كما هو موضح في الشكل ]



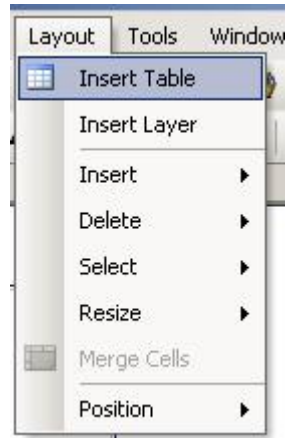
الشكل ٤ [في الـ (User Control) نقوم بعمل الشكل التالي "وظيفة الأداة هو البحث بواسطة جول"]

```
protected void btnSearch_Click(object sender, EventArgs e)
{
    Response.Redirect(string.Format(
        "http://www.google.com/search?q={0}", txtQuery.Text));
}
```

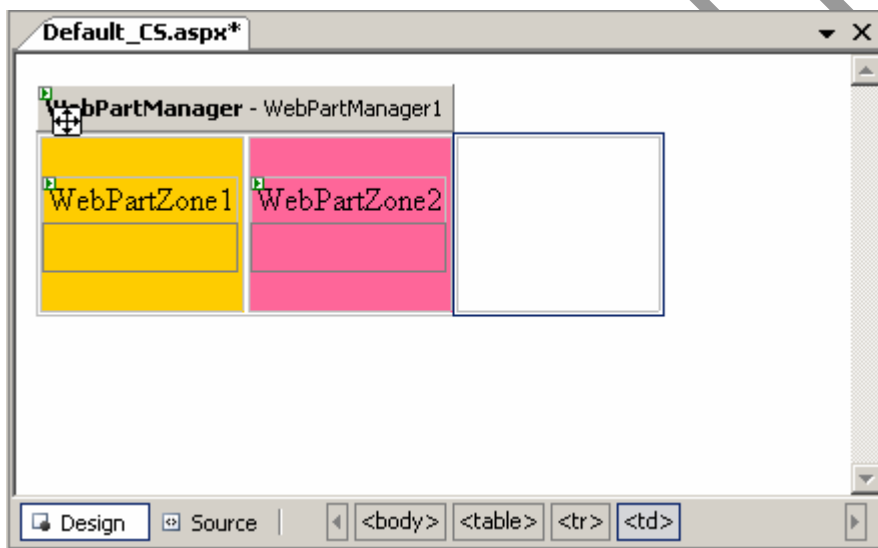
الكود بالـ C#

```
Protected Sub btnSearch_Click(ByVal sender As Object, _
    ByVal e As System.EventArgs) Handles btnSearch.Click
    Response.Redirect(String.Format( _
        "http://www.google.com/search?q={0}", txtQuery.Text))
End Sub
```

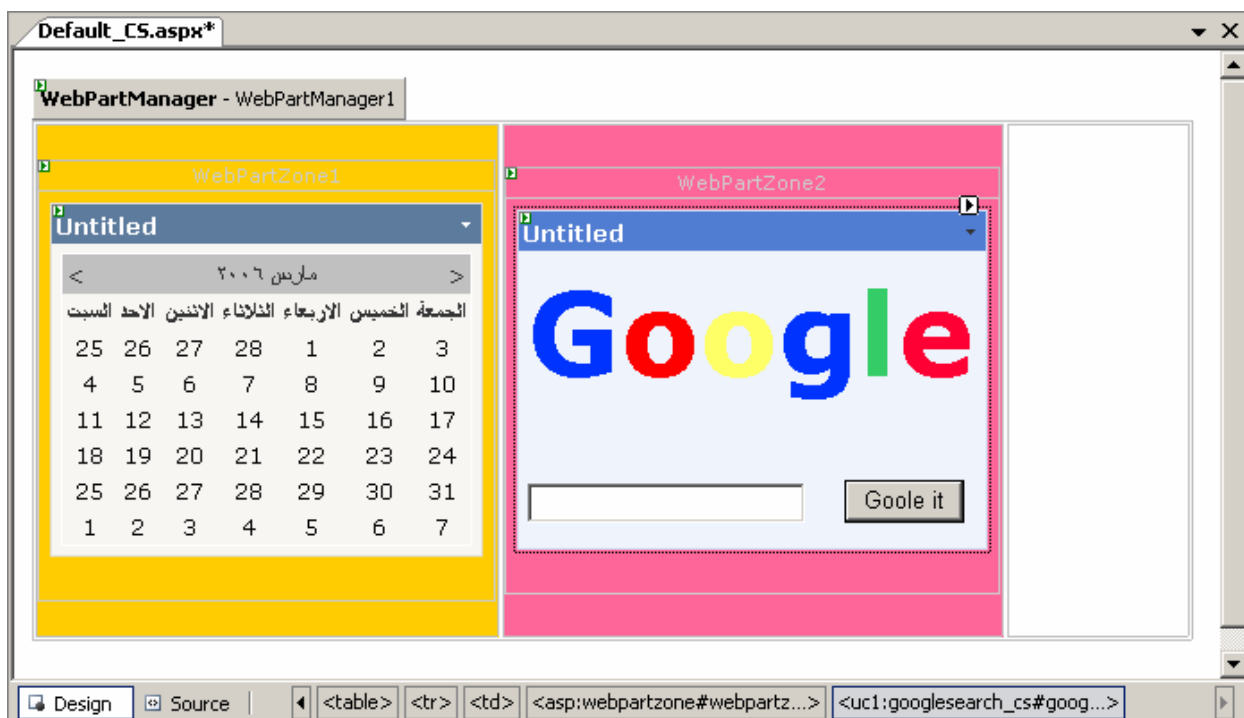
الكود بالـ VB.net



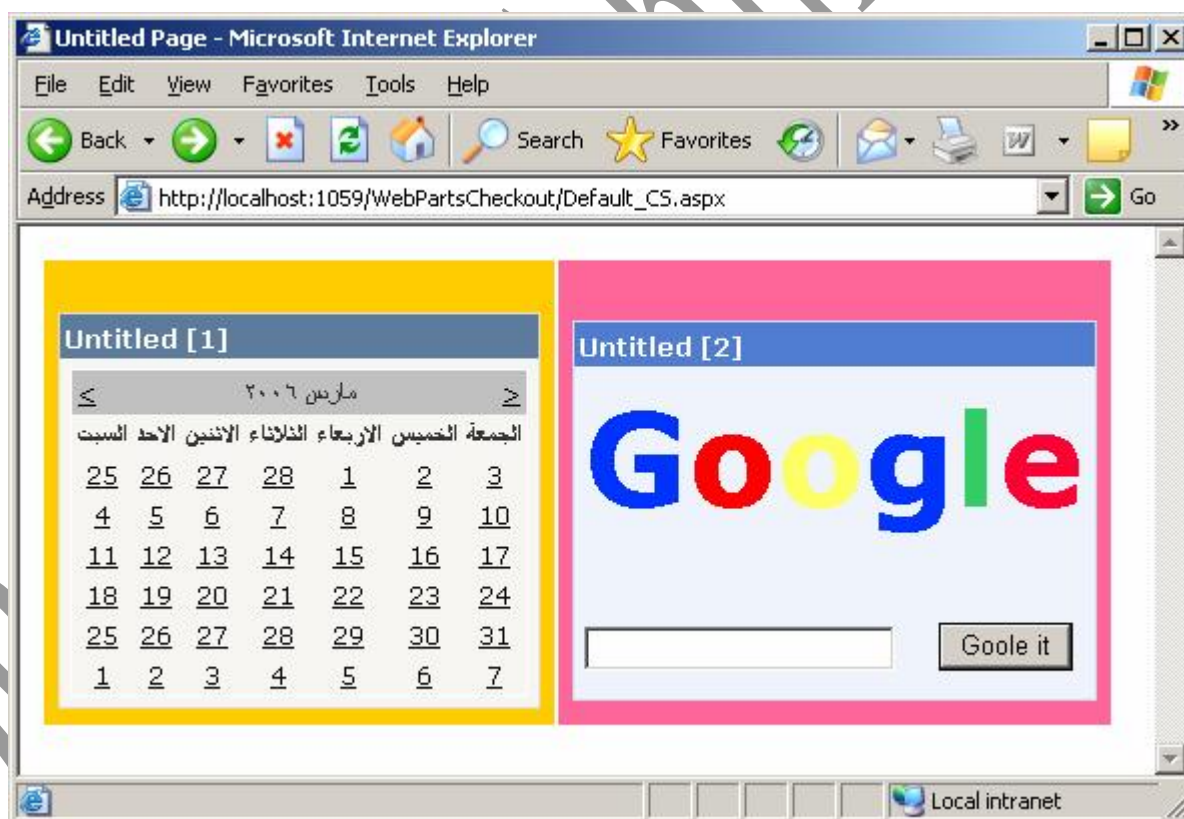
الشكل ٥ [ نقوم بأضافة جدول صفحة ]



الشكل 6 [ نقوم بأضافة كلاً من الأداة (WebPartManager) و الـ (WebPartZone) ]



الشكل 7 [ نقوم بإضافة الأدوات المراد إضافتها داخل الـ (WebPartZone) ، نقوم بإضافة أداة النتيجة ، ثم الـ (UserControl) ]



الشكل 8 [ هو الشكل الذي سيظهر عن تشغيل المثال ، ولكن كما نلاحظ أن العنوان مكتوب (Untitled) ولتغييره ادخل على الـ

[ (source view)

"ججججوجل مع بحث" ID="GoogleSearch\_CS1" title="ججججوجل مع بحث" <uc1:GoogleSearch\_CS



```
runat="server" />
```

Title=" :D النتيجة"

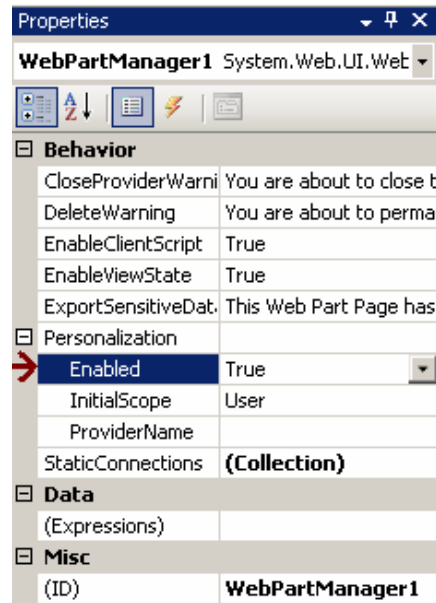
الشكل 9 [ أضف التوصيف ( Title ) سوف تلاحظ رفض الـ (VS) للتعديل الذي قمت به وسيتم إيضاح كل هذا لاحقاً ]

title=" بحث مع ججججوجل"

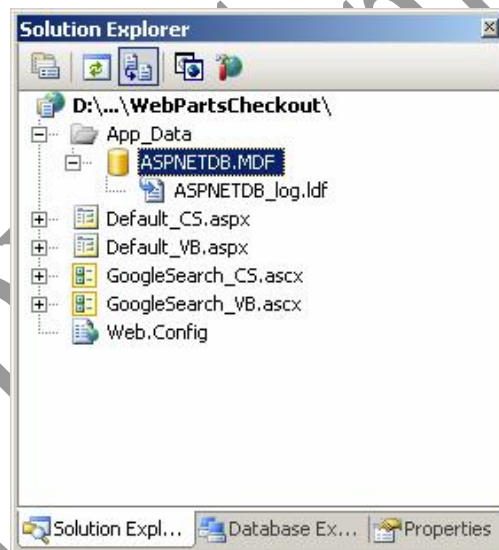
الشكل ١٠



الشكل 11 [ هو الشكل الجديد بعد تعديل العنوان ، و لكن حتى الآن لم نستفيد من مميزات الـ (WebParts) ، و الآن سوف نوضح كيفية التعامل مع الـ (WebParts) و الاستفادة من مميزات الجبارة في الـ (Personalisation) ]

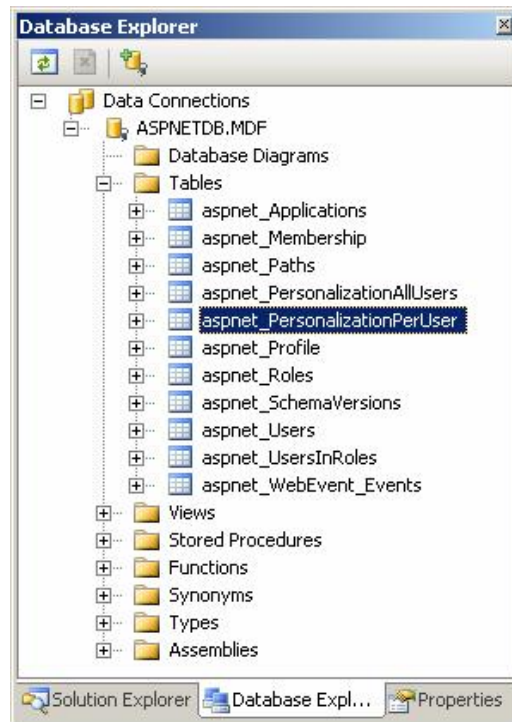


الشكل 12 [ قم بتنشيط الـ (Personalization) إلى (True) ؛ سوف يتطلب تفعيل هذه الخاصية إلى وجود قاعدة بيانات الـ (Membership and personalization) كما هو موضح في الشكل التالي [13] ]

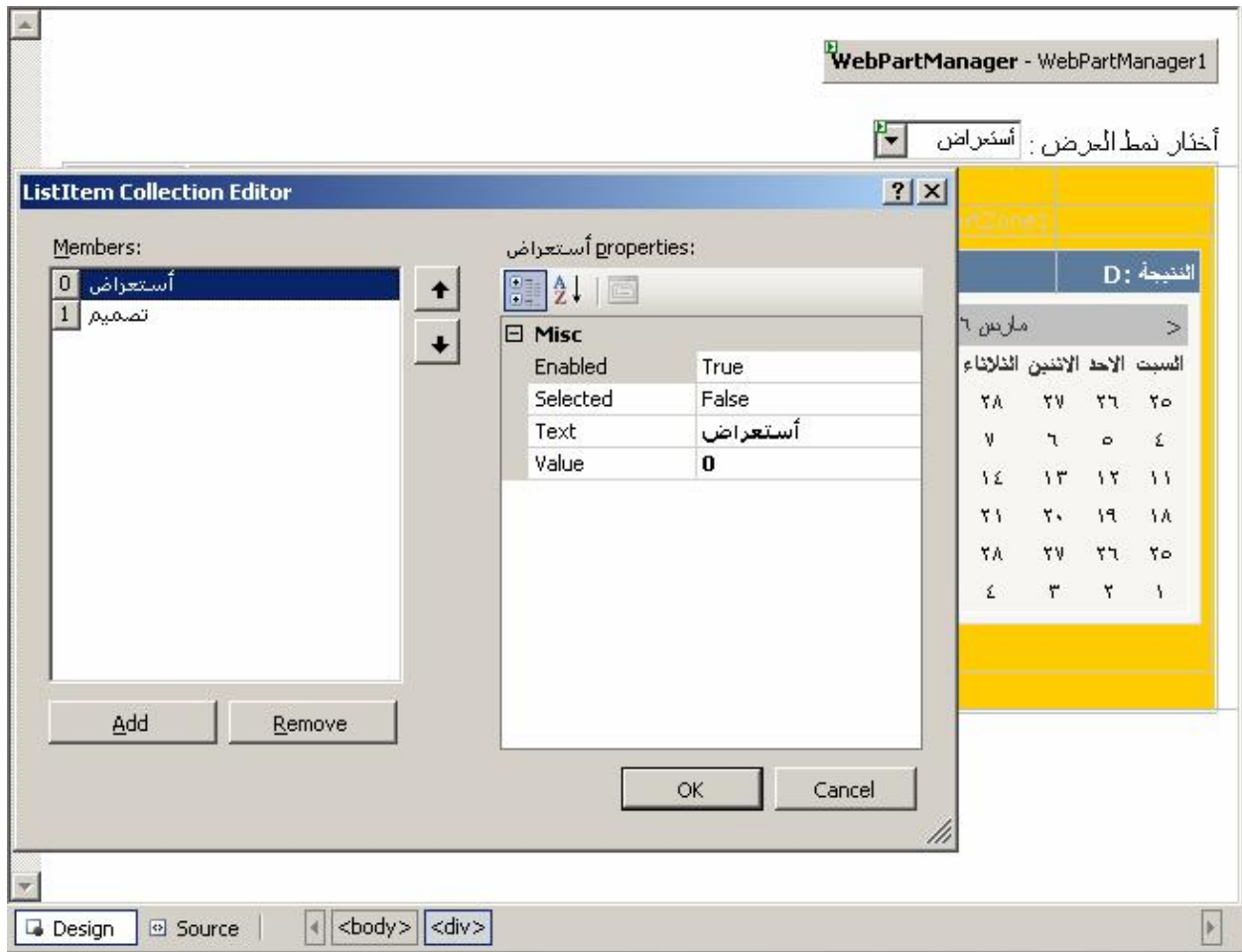


الشكل ١٣ [ هي قاعدة البيانات التي تقوم بإدارة كل إمكانيات الـ (Personalization) و الـ (Membership) ]





الشكل 14 ] سوف تلاحظ أن معظم أمكانيات الـ (SQL server) موجودة في الـ (VS.net) ؛ بسبب أن الـ (VS) يتكامل مع الـ [ (MS-SQL Server)



الشكل ١٥] قم بإضافة قائمة منزلقة ، و أضف فيها عنصران كما هو موضح - أستعراض = ٠ / تصميم = ١ - ، و قم بتنشيط

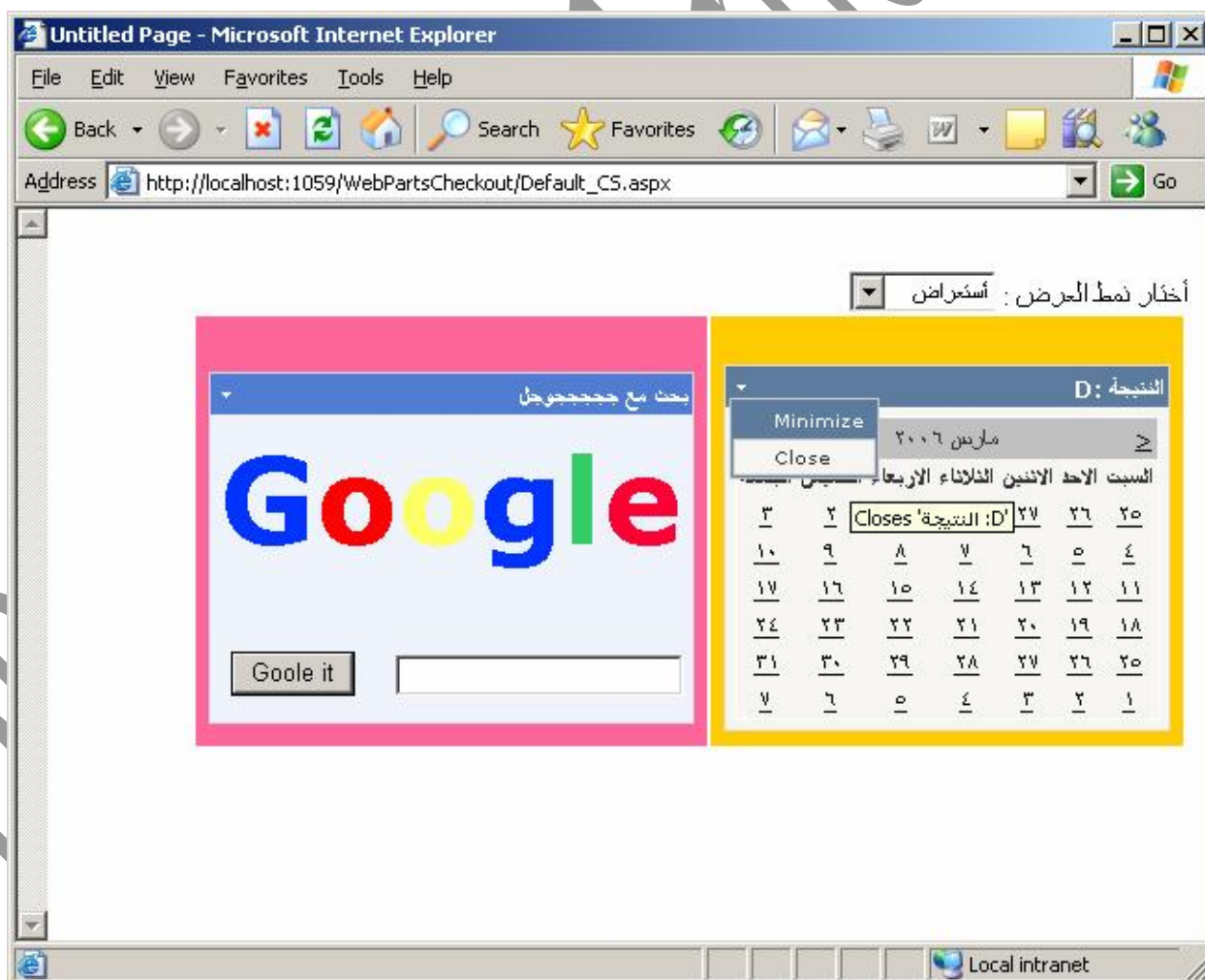
(Autopostback) إلى (True) للقائمة المنزلقة ، ثم أضف الكود التالي]

١- الكود بالـ (C#)

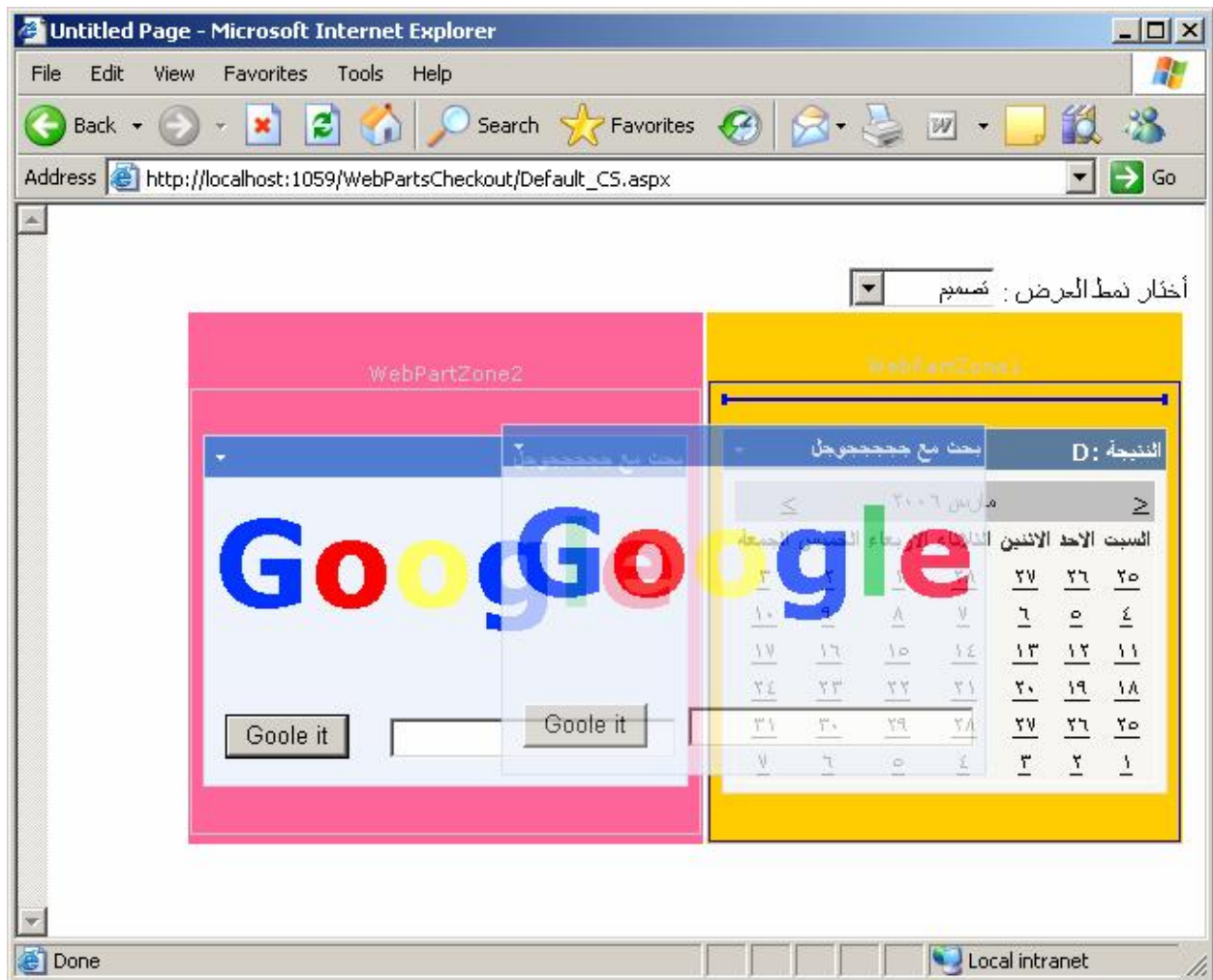
```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    switch (int.Parse(ddl_ShowType.SelectedValue))
    {
        case 0:
            WebPartManager1.DisplayMode = WebPartManager.BrowseDisplayMode;
            break;
        case 1:
            WebPartManager1.DisplayMode = WebPartManager.DesignDisplayMode;
            break;
        default:
            WebPartManager1.DisplayMode = WebPartManager.BrowseDisplayMode;
            break;
    }
}
```

```
Protected Sub DropDownList1_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles ddl_ShowType.SelectedIndexChanged
    Select Case Val(ddl_ShowType.SelectedValue)
        Case 0
            WebPartManager1.DisplayMode =
WebPartManager.BrowseDisplayMode
        Case 1
            WebPartManager1.DisplayMode =
WebPartManager.DesignDisplayMode
        Case Else
            WebPartManager1.DisplayMode =
WebPartManager.BrowseDisplayMode
    End Select
End Sub
```

٢- الكود بالـ (VB.net)



الشكل 16 [ هو الشكل عند الانتهاء من المثال ، يمكنك الآن أن تقوم بإغلاق أو تصغير محتويات الـ (WebPartZone) ]



الشكل 17 [يوضح إمكانية تحريك العناصر]



الشكل 18 [يوضح سهولة إمكانية نقل العناصر من الـ ( WebPartZone ) إلى الأخرى دون مشاكل]





الشكل 19 [بالإضافة الى أن التطبيق يقوم بحفظ إعدادات المستخدم ، قم بإغلاق العناصر الموجودة]

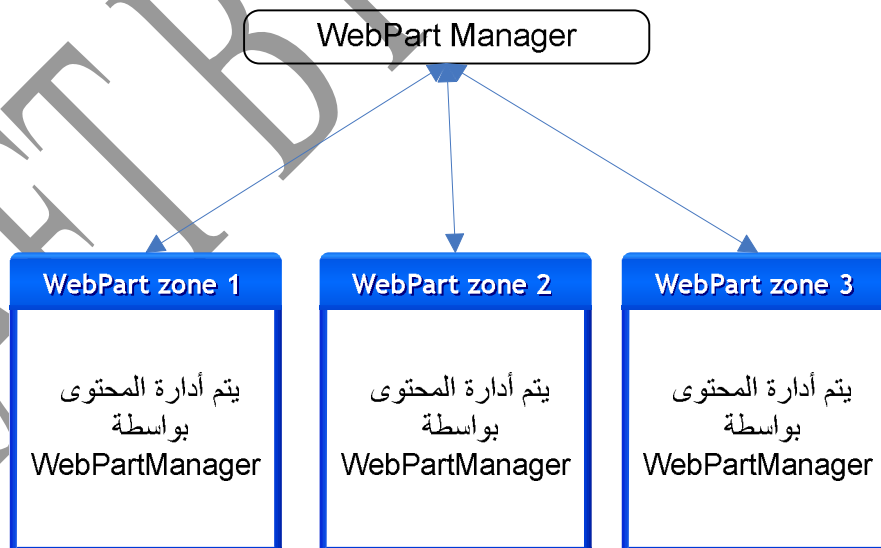


الشكل 20 [ مشكلة :- و هي كيفية إظهار المكونات مرة أخرى ]

Start Page	aspnet_Person...ASPNETDB.MDF)	aspnet_Perso...ASPNETDB.MDF)	Default_CS.aspx.cs	Default_CS.aspx
Id	PathId	UserId	PageSettings	LastUpdatedDate
7927a16b-f385-41a8-92b7-910a0f5b593c	f65571e9-785c-4252-94f5-6d446f0954f9	96374a04-d5c6-4bbf-a747-fbc1cc663cfa	<Binary data>	12/03/2006 11:06:57 ص
*	NULL	NULL	NULL	NULL

الشكل 21 [ حل المشكلة :- الموضوع بسيط للغاية هناك طريقان لحل هذه المشكلة ؛ ١- إلغاء السطر الخاص بالمستخدم من قاعدة البيانات - لا أحبذ هذا الحل- ، ٢- إستعمال مكونات إضافية من الـ (Toolbar) لحل هذه المشكلة ]

الآن لنشرح ماذا يحدث بالضبط في هذه العملية :-



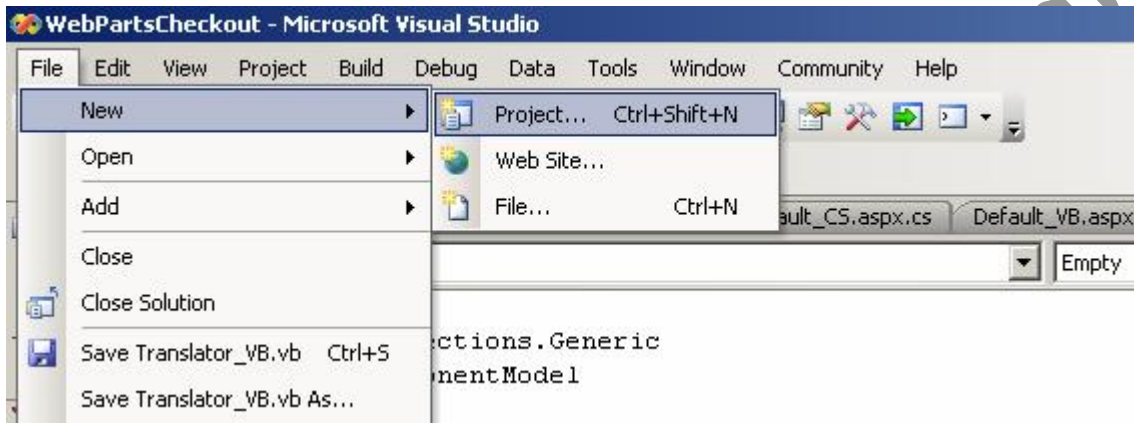
الشكل 22 [ كما هو موضح في الشكل أن عملية إدارة المحتوى - المحتوى هنا تعني الأدوات الموجودة بداخل الـ (WebPart Zone) - تتم بواسطة الـ (WebPart Manager) ]



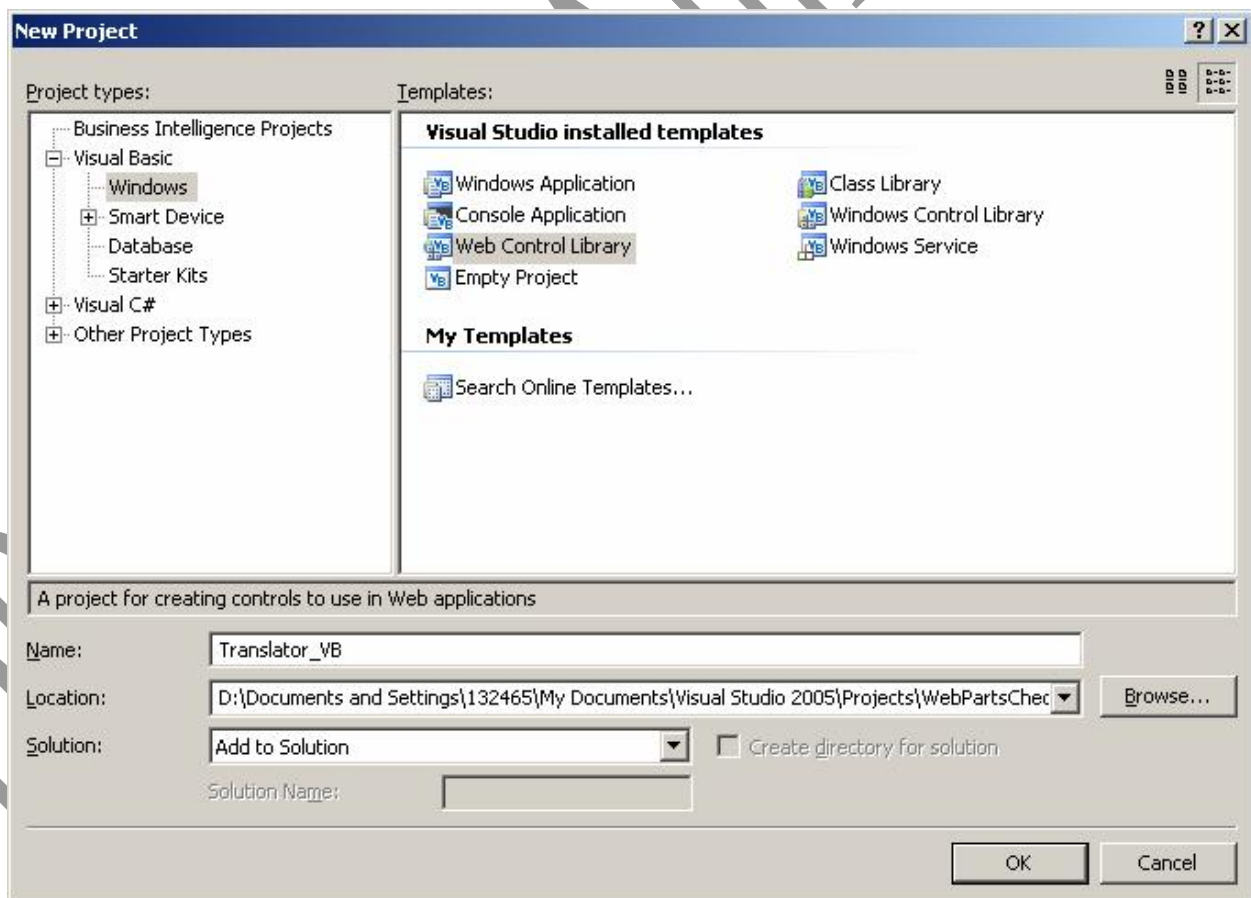
- لاحظنا أننا قمنا بإضافة معرف بالاسم (Title) لتغيير اسم الـ (WebPart Zone) ؛ ولابد من إيضاح شئ بسيط جدا  
الا وهو ان الـ (WebPart Zone) تُكسب ما بداخلها من أدوات بعض الصفات الجديدة و التي هي في الأصل موروثه  
من (WebPart Generic) .

- الآن لنبدأ بعمل بعض الإضافات الجديدة على المثال و هي أن نقوم بوراثه صفات المكتبة (WebPart) لنكسب بعض  
الأمكانيات التي يمكن إضافتها لصفحات التطبيق.

و الآن الى المثال و هو عبارة عن (Custom control) يقوم بوراثه المكتبة (WebPart) و الآن لنبدأ العمل.

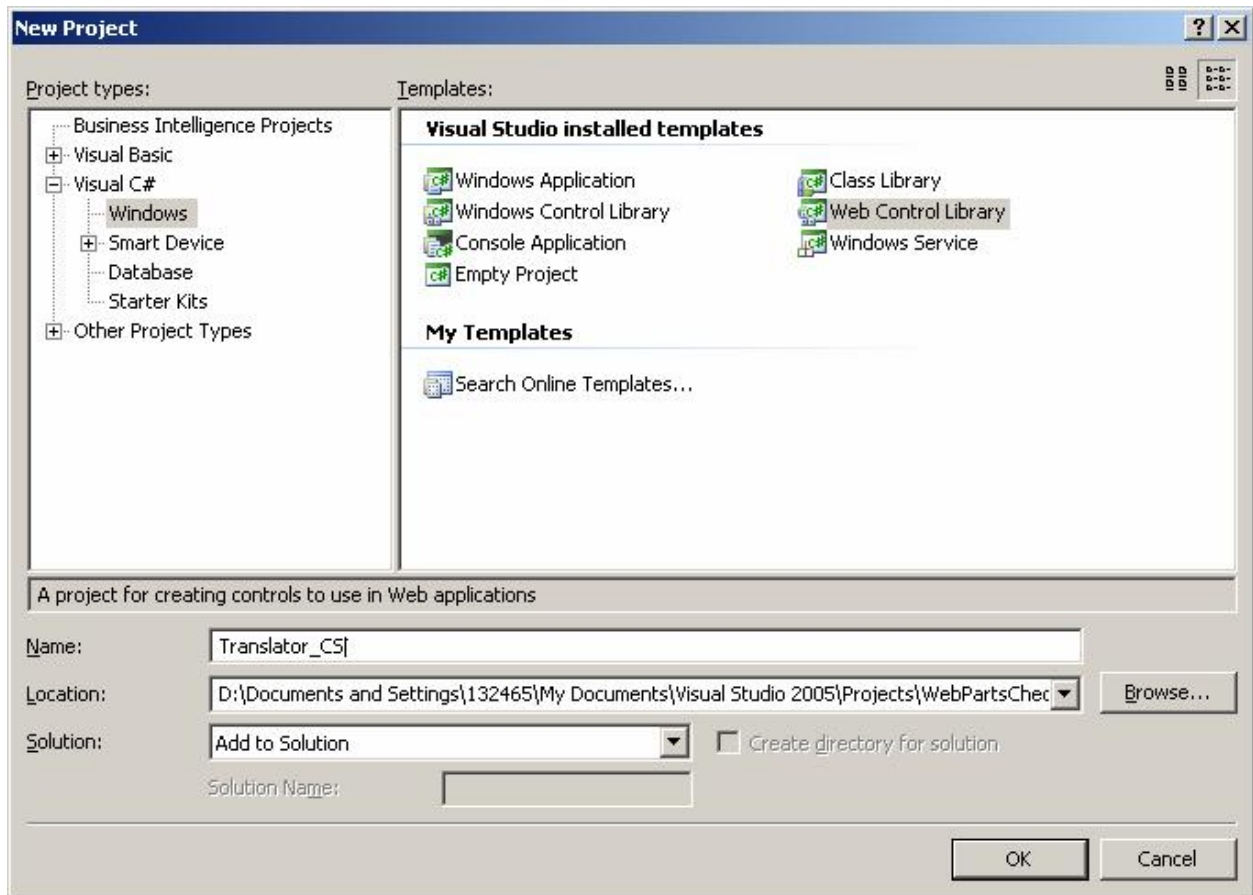


الشكل 23] سوف نقوم بإضافة مشروع جديد من القائمة (File à Project)

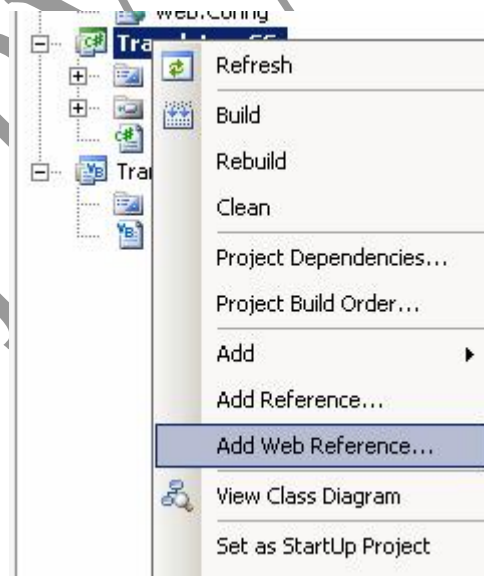


الشكل 24] هذا الشكل لمبرمجي الـ (VB) ، أختار اللغة المراد التعامل معها و بعد ذلك نقوم بأختيار نوع المشروع وهو

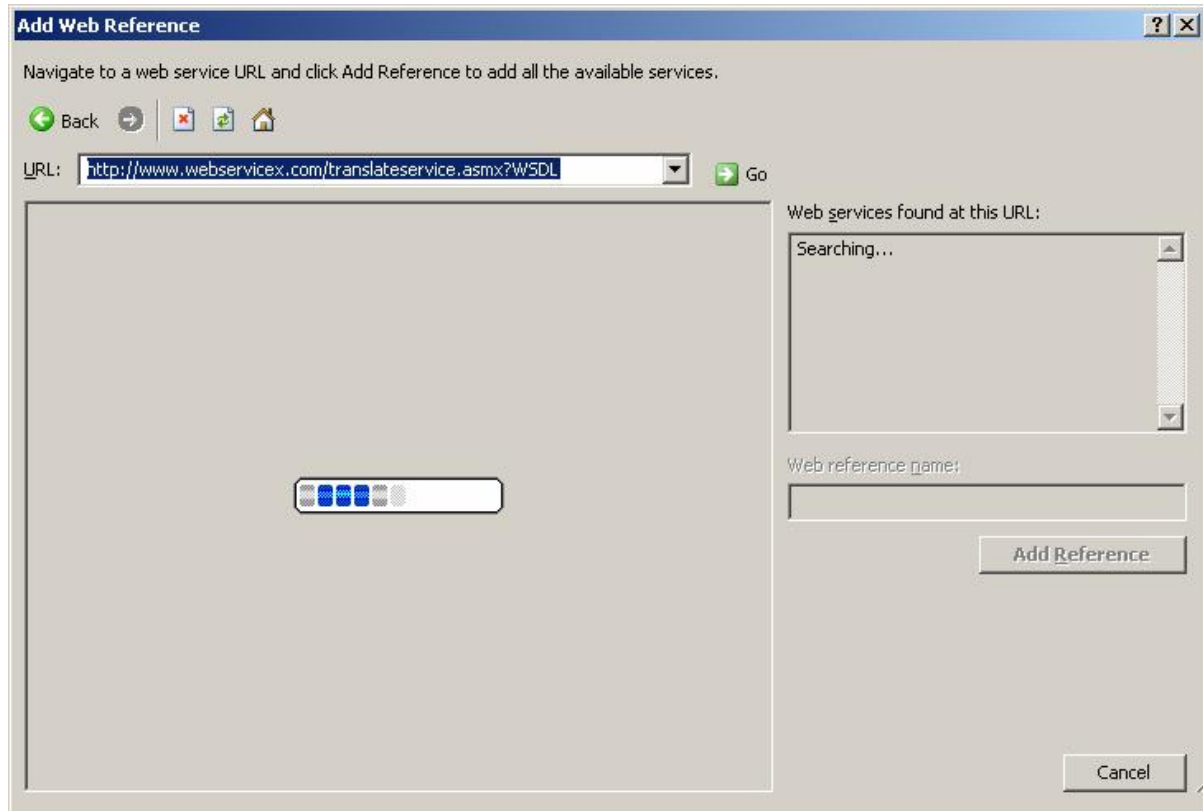
[ (Web Control library)



الشكل 25] هذا الشكل لمبرمجي الـ (C#) [

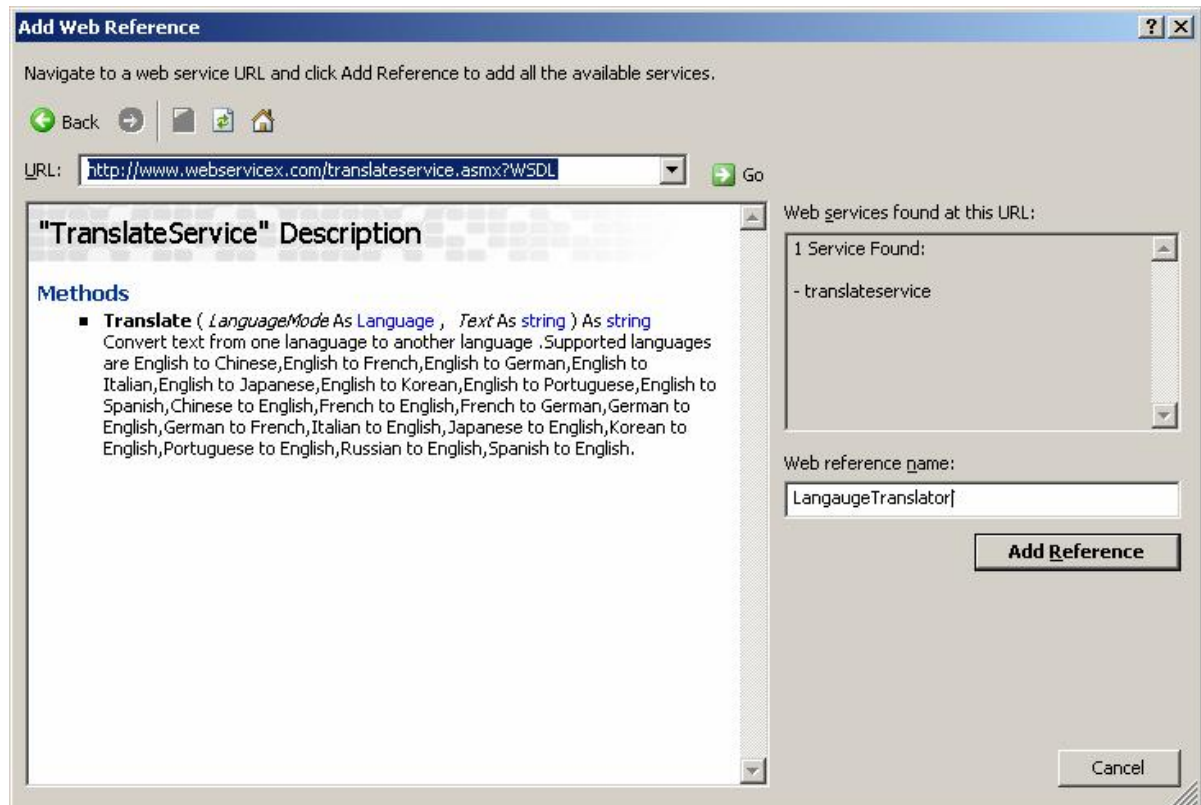


الشكل 26 ] يوضح المثال الذي سنطبقه ، وهو عبارة عن مترجم من اللغة الإنجليزية إلى اليابانية باستخدام خدمات الويب (WebService) لهذا سوف نحتاج إلى مرجع ويب (Web Reference) لمزود الخدمة و كالعادة سوف يقوم الـ (Visual studio) بإنشاء المكتبة الوسيطة (Proxy class) التي تسهل التعامل مع خدمة الويب (WebService) المراد التعامل معها و أول خطوة في هذا الموضوع هو النقر بالزر الأيمن للماوس و إختيار (Add Web Reference)

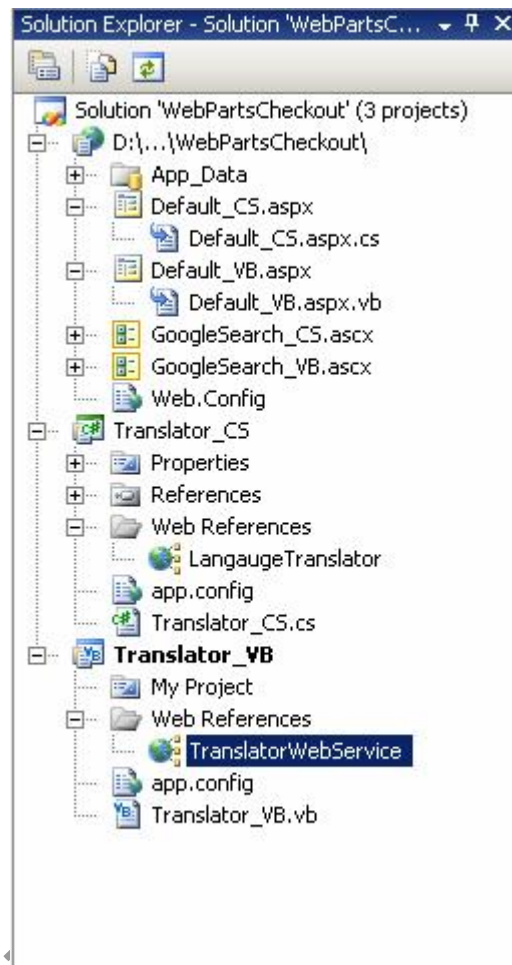


الشكل 27 ] سوف تظهر هذه النافذة (dd Web Reference) وسوف نضع عنوان الخدمة المراد التعامل معها و هي

<http://www.websvcex.com/translateService.asmx?WSDL> ]



الشكل 28] عند تحميل الخدمة سوف يظهر لك نبذة عن الخدمة و المتغيرات التي تحتاجها و كل ما يخص هذه الخدمة]



الشكل ٢٩ [ هذا هو الشكل بعد اضافة الخدمة (Web Service) ]

```
Imports System
Imports System.Collections.Generic
Imports System.ComponentModel
Imports System.Text
Imports System.Web
Imports System.Web.UI.WebControls.WebParts
Imports System.Web.UI
Imports System.Web.UI.WebControls
```

```
<DefaultProperty("Text")> _
```

```
Public Class Translator_VB
```

```
Inherits WebPart
```

```
Private pStrText As String = Nothing
```

```
Private txtStringToTranslate As TextBox
```

فى هذا السطر نقوم بوراثه المكتبة (WebPart)

متغير من نوع سلسلة حرفيه و التى سوف تحمل الكلمة المراد ترجمتها

متغير من نوع حقل و الذي سوف نقوم بعرضه و الذي سوف يكون مدخل الكلمات المراد ترجمتها

```
Private lblTranslatedString As Label
```

عارض نص (Label) و الذي سيتم به عرض النتيجة

```
Public Sub New()
```

دالة المنشئ (Constructor) و هنا يتم تفعيل خاصية القابلية للأغلاق

```
Me.AllowClose = True
```

```
End Sub
```

<Personalizable(), WebBrowsable()> \_

هنا يتم منح و إتاحة الخاصيتين :- القابلية للتخصيص الشخصي (Peronalization) و القابلية للاستعراض (WebBrowsable)

```
Public Property strToTranslate() As String
```

إضافة خاصية للمكتبه و هي الكلمة التي سوف يتم ترجمتها ؛ و ترجع أهمية هذه الخاصية لإتاحة واجهة برمجية للتعامل مع المكتبه ؛ فكما نلاحظ لا يمكن التعامل مع النص أو حقل النص مباشرة

```
Get
```

```
Return pStrText
```

```
End Get
```

```
Set(ByVal value As String)
```

```
pStrText = value
```

```
End Set
```

```
End Property
```

```
Protected Overrides Sub CreateChildControls()
```

```
Controls.Clear()
```

مسح كل الأدوات الموجودة مسبقاً

```
txtStringToTranslate = New TextBox()
```

إنشاء نسخة جديدة من حقل النص فارغة

```
txtStringToTranslate.Text = Me.strToTranslate
```

نسخ محتويات الخاصية (strToTranslate) إلى حقل النص كقيمة ابتدائية

```
Me.Controls.Add(txtStringToTranslate)
```

تثبيت حقل النص على الصفحة

```
Dim btnTranslate As New Button()
```

إنشاء زر جديد و الذي سيتولى الترجمة عند الضغط عليه

```
btnTranslate.Text = "ترجم"
```

تغيير نص الزر إلى اسم ملانم

```
AddHandler btnTranslate.Click, AddressOf Me.btnTranslate_Click
```

إنشاء مناول الحدث (Event Handler) لكي يقوم بتنفيذ مهمه معينة -التعليمات الموجودة في الدالة (btnTranslate\_click)- عند الضغط على الزر

```
Me.Controls.Add(btnTranslate)
```

```
lblTranslatedString = New Label()  
  
lblTranslatedString.BackColor = System.Drawing.Color.Yellow  
  
Me.Controls.Add(lblTranslatedString)  
  
ChildControlsCreated = True  
  
End Sub  
  
Private Sub btnTranslate_Click(ByVal sender As Object, _  
                                ByVal e As EventArgs)  
  
    If txtStringToTranslate.Text <> String.Empty Then  
  
        Me.strToTranslate = txtStringToTranslate.Text  
  
        txtStringToTranslate.Text = String.Empty  
  
        Dim ws As New TranslatorWebService.TranslateService  
  
        lblTranslatedString.Text = "<br/>" & Me.strToTranslate & "-->" & _  
        ws.Translate(TranslatorWebService.Language.EnglishTOJapanese, _  
        Me.strToTranslate)  
  
    End If  
End Sub  
End Class
```

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Text;  
using System.Web;  
using System.Web.UI;  
using System.Web.UI.WebControls;
```



```
using System.Web.UI.WebControls.WebParts;
```

```
namespace Translator_CS
```

```
{
```

```
[DefaultProperty("Text")]
```

```
public class Translator_CS : WebPart
```

في هذا السطر نقوم بوراثة المكتبة (WebPart)

```
{
```

```
private string pStrText = "";
```

متغير من نوع سلسلة حرفية و التي سوف تحمل الكلمة المراد ترجمتها

```
private TextBox txtStringToTranslate;
```

متغير من نوع حقل و الذي سوف نقوم بعرضه و الذي سوف يكون مدخل الكلمات المراد ترجمتها

```
private Label lblTranslatedString;
```

عارض نص (Label) و الذي سيتم به عرض النتيجة

```
public Translator_CS()
```

```
{
```

```
this.AllowClose = true;
```

دالة المنشئ (Constructor) و هنا يتم تفعيل خاصية القابلية للإغلاق

```
}
```

```
[Personalizable(), WebBrowsable()]
```

هنا يتم منح وإتاحة الخاصيتين :- القابلية للتخصيص الشخصي (Peronalization) و القابلية للاستعراض (WebBrowsable)

```
public string strToTranslate
```

إضافة خاصية للمكتبه و هي الكلمة التي سوف يتم ترجمتها ؛ وترجع أهمية هذه الخاصية لإتاحة واجهة برمجية للتعامل مع المكتبه ؛ فكما

نلاحظ لا يمكن التعامل مع النص أو حقل النص مباشرة

```
{
```

```
get
```

```
{
```

```
return pStrText;
```

```
}
```

```
set
```

```
{
```

```
pStrText = value;
```

```
}
```

```
}
```

```
protected override void CreateChildControls()
```

```
{
```

```
Controls.Clear();
```

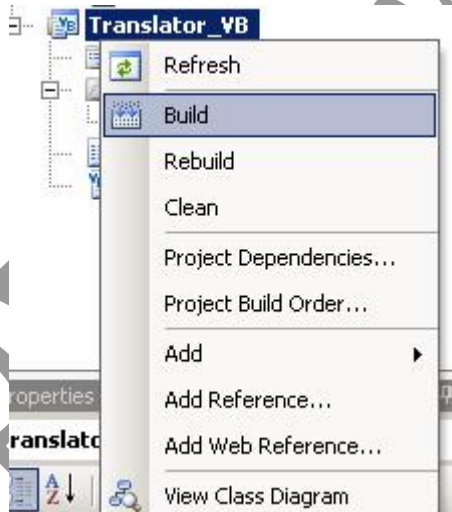
مسح كل الأدوات الموجودة مسبقاً

```
txtStringToTranslate = new TextBox();  
إنشاء نسخة جديدة من حقل النص فارغة  
txtStringToTranslate.Text = this.strToTranslate;  
نسخ محتويات الخاصية (strToTranslate) إلى حقل النص كقيمة ابتدائية  
this.Controls.Add(txtStringToTranslate);  
تثبيت حقل النص على الصفحة  
  
Button btnTranslate = new Button();  
إنشاء زر جديد والذي سيتولى الترجمة عند الضغط عليه  
btnTranslate.Text = "Translate";  
تغيير نص الزر إلى اسم ملائم  
btnTranslate.Click += new EventHandler(btnTranslate_Click);  
إنشاء مناول الحدث (Event Handler) لكي يقوم بتنفيذ مهمه معينة -التعليمات الموجودة في الدالة (btnTranslate_Click)- عند الضغط  
على الزر  
this.Controls.Add(btnTranslate);  
تثبيت الزر على النافذة  
  
lblTranslatedString = new Label();  
إنشاء أداة نص  
lblTranslatedString.BackColor = System.Drawing.Color.Yellow;  
تغيير لون الخلفيه إلى اللون الأصفر  
this.Controls.Add(lblTranslatedString);  
تثبيت أداة النص على الصفحة  
ChildControlsCreated = true;  
تأكيد إنشاء الأدوات  
}  
  
private void btnTranslate_Click(Object sender, EventArgs e)  
إنشاء الدالة التي سوف تنفذ عند ضغط المستخدم على الزر  
{  
if (txtStringToTranslate.Text != String.Empty )  
التأكد من عدم حل النص من أى كلمه  
{  
this.strToTranslate = txtStringToTranslate.Text;  
إحلال قيمة الخاصية (trToTranslate) بقيمة حقل النص  
txtStringToTranslate.Text = String.Empty;  
تفريغ حقل النص  
  
LangaugeTranslator.TranslateService Trans = new  
global::Translator_CS.LangaugeTranslator.TranslateService();
```

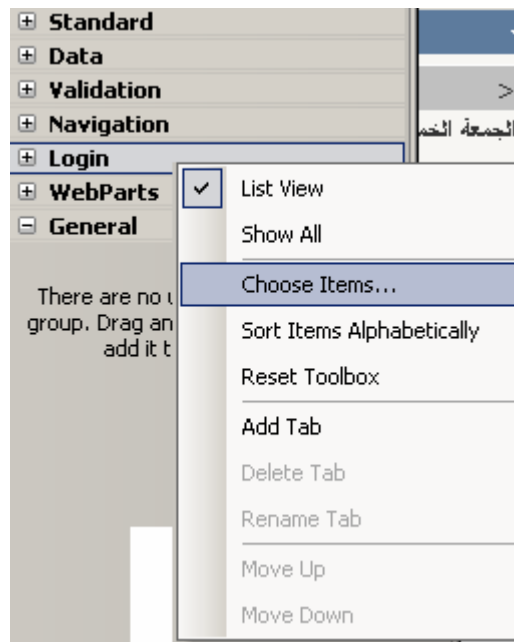
إنشاء متغير من نوع الخدمة ( Web Service )

```
lblTranslatedString.Text = "<br/>" + this.strToTranslate + "-->"
+
Trans.Translate(global::Translator_CS.LangaugeTranslator.Language.EnglishTOJapanese,this.strToTranslate);
}
}
}
}
```

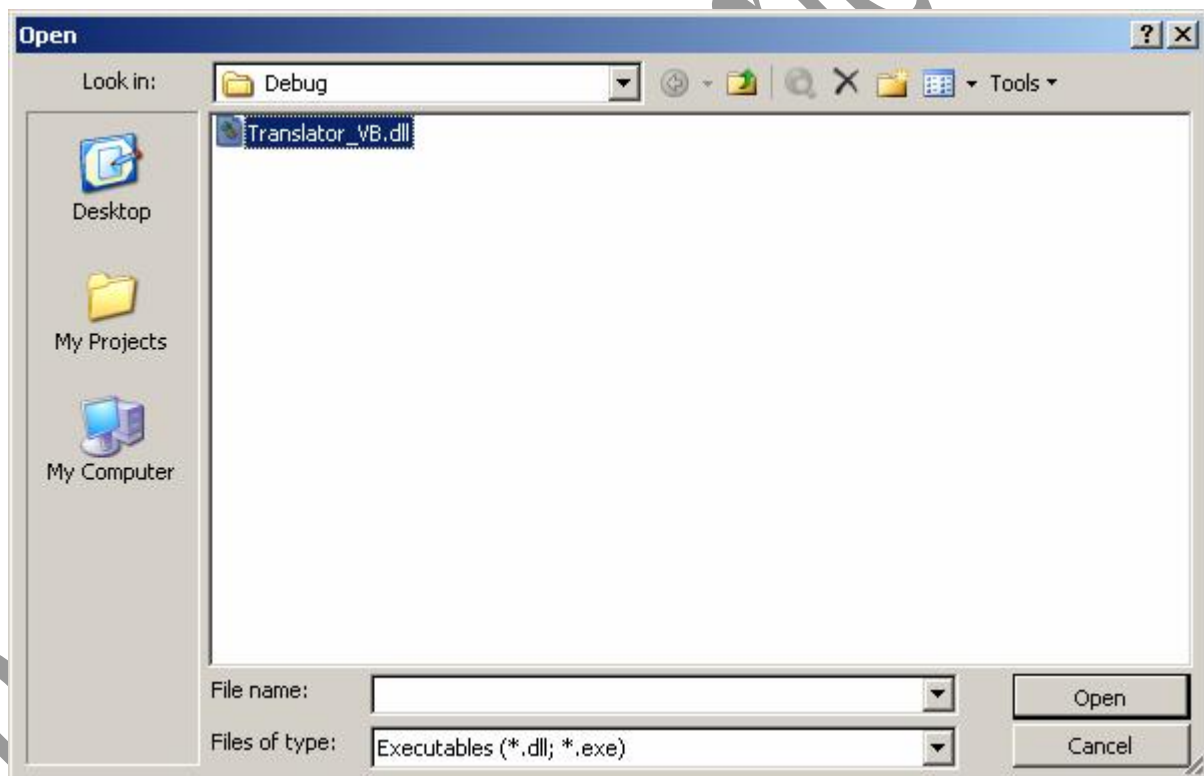
تحديد اللغة المراد الترجمة منها وإليها وتحديد الكلمة المراد ترجمتها



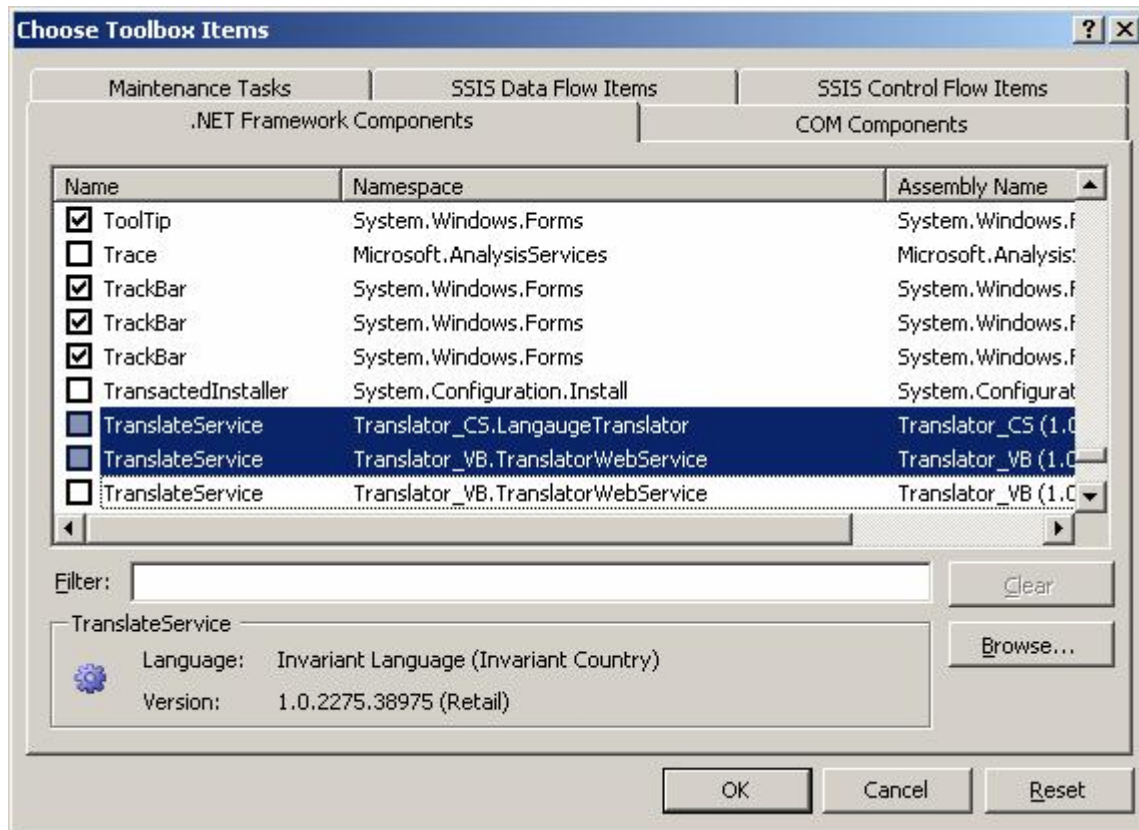
الشكل 30 [ الآن و بعد كتابة الكود و الإنتهاء من كل المطلوب بالنسبة للأداة نقوم بترجمتها ]



الشكل 31 بعد ذلك نعود للمشروع مرة أخرى و نقوم بإضافة الأداة التي قمنا بتصميمها الآن [



الشكل 32 [ الآن نختار الملف الذي يحتوي على الأداة و هو موجود في مجلد ( bin\debug ) داخل مجلد مشروع المكتبة التي تم إنشاؤها ]



الشكل 33 [سوف تظهر المكتبة - كما نرى - فنقوم بالإختيار ثم الضغط (OK)]



الشكل 34 [ هو شكل الأدوات بعد إضافتها إلى صندوق الأدوات (Tool Box)]



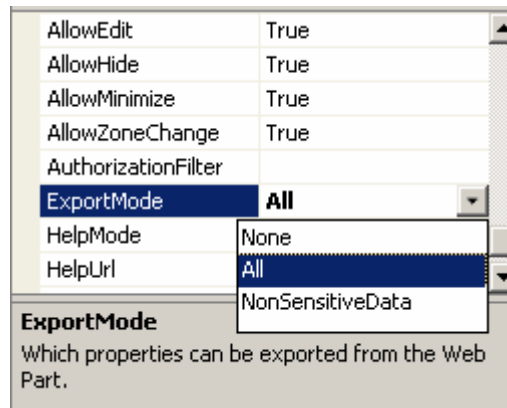
الشكل 35 [ هذا هو الشكل في التصميم و بعد اضافة (WebZone) جديدة تحتوى على الاداة الخاصة بنا ]



الشكل 36 [ هو الشكل عند التنفيذ ]

- هناك خاصية جديدة و مميزة - من وجهة نظري - وهى إمكانية تمكين المستخدم من استيراد و تصدير خصائص الأداة .

و الشكل التالي [37] يوضح كيفية تشغيل هذه الإمكانية :-



الشكل 37

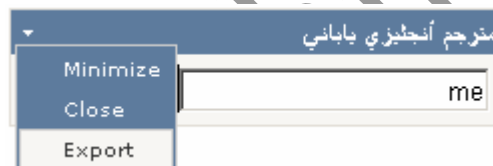
ونلاحظ في الشكل [37] وجود ثلاث خيارات و هم :-

- ١- (None) : ومعناها عدم إظهار الخاصية عند المستخدم .
- ٢- ( ALL ) : ومعناها إستيراد و تصدير كل الخصائص .
- و لكن ماذا لو كانت الأداة تحمل بيانات سرية و خاصة عن المستخدم هنا يأتي الخيار الثالث والأخير
- ٣- (NonSensitiveData) : وهو يعتبر الخيار الأمثل لنا في هذه الحالة
- هناك شئ آخر وهو يجب أن يتم تفعيل هذه الخاصية على مستوى التطبيق من خلال تفعيل الخاصية (EnableExport) في ( WebPart ) في ملف (Web.config)

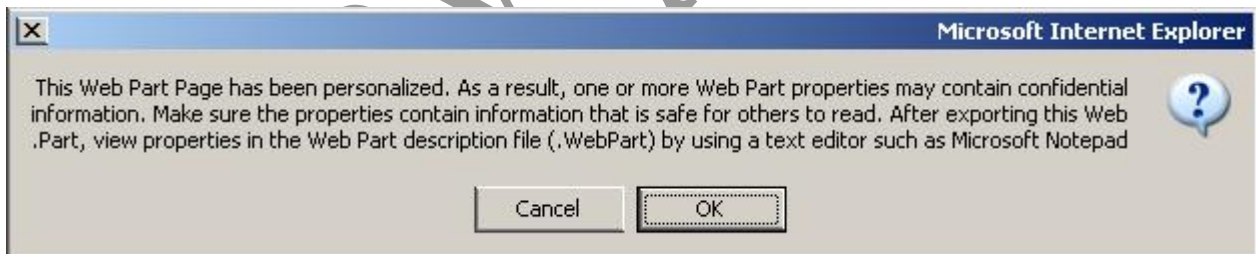
```
<!-->
<system.web>
  <webParts enableExport="true"/>
  <!--
```

الشكل 38

- الآن لنرى كيف سيتعامل المستخدم مع هذه الخاصية (EnableExport) - وكيف يستفيد منها



الشكل ٣٩ أول خطوة هي من قائمة الـ (WebPart Zone) سوف نلاحظ ظهور الخيار (Export) [



الشكل 40 رسالة تحذيرية لوجود بيانات مهمة و حساسة و كيفية أمكانية عرض الملف [





الشكل 41[الأن نقوم بحفظ ملف الإعدادات التي تم تصديرها من الأداة]

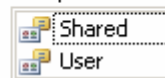
```
<?xml version="1.0" encoding="utf-8"?>
<webParts>
  <webPart xmlns="http://schemas.microsoft.com/WebPart/v3">
    <metaData>
      <type name="Translator_CS.Translator_CS" />
      <importErrorMessage>Cannot import this Web Part.</importErrorMessage>
    </metaData>
    <data>
      <properties>
        <property name="AllowClose" type="bool">True</property>
        <property name="Width" type="unit" />
        <property name="strToTranslate" type="string">me</property>
        <property name="AllowMinimize" type="bool">True</property>
        <property name="AllowConnect" type="bool">True</property>
        <property name="ChromeType" type="chrometype">Default</property>
        <property name="TitleIconImageUrl" type="string" />
        <property name="Description" type="string" />
        <property name="Hidden" type="bool">False</property>
        <property name="TitleUrl" type="string" />
        <property name="AllowEdit" type="bool">True</property>
        <property name="Height" type="unit" />
        <property name="HelpUrl" type="string" />
        <property name="Title" type="string">ياباني أنجليزي مترجم</property>
        <property name="CatalogIconImageUrl" type="string" />
        <property name="Direction" type="direction">NotSet</property>
        <property name="ChromeState" type="chromestate">Normal</property>
      </properties>
    </data>
  </webPart>
</webParts>
```

```
<property name="AllowZoneChange" type="bool">True</property>
<property name="AllowHide" type="bool">True</property>
<property name="HelpMode" type="helpmode">Navigate</property>
<property name="ExportMode" type="exportmode">All</property>
</properties>
</data>
</webPart>
</webParts>
```

- هذه هي محتويات الملف و كما نلاحظ انها عبارة عن ملف (XML) و به الخواص

رائع جدا و لكن كيف يمكننا إستيراد خواص الأداة الخاصة بنا !!! ..... هذا ما سوف نتحدث عنه لاحقا.  
هناك أيضا أجزاء كثيرة نحتاج الى معرفتها لكل مكسب إمكانيات أكثر.  
قد تلاحظون عند تطبيق المثال أنه عند البحث تختفى كلمة البحث !!! .... فماذا لو أردنا ظهور هذه الكلمة مع المستخدم  
!!!! ..... ؛ في هذه الحالة سيتم الإعتماد على الـ (Personalization) بشكل قوى جدا ؛ وبالتالي سنحتاج إلى إجراء  
تعديل كود بسيط في مكتبة الأداة الخاصة بالترجمة لنقوم بتلبية هذا الغرض ؛ حيث فيها سنقوم بتحديد نطاق التخصيص  
(Personalization Scope) و كما نرى في الشكل التالي [42] كيف يمكننا ذلك

```
}
[Personalizable(PersonalizationScope.), WebBrowsable
public string strToTranslate
{
```



الشكل ٤٢

وسوف نلاحظ وجود نوعان منها و هو مشترك (Shared) و مخصص لمستخدم (User) و الفرق بينهم :-  
١- الأول (Shared) عند إستخدامه يتم حفظ آخر كلمة لكل التطبيق بمعنى أنه إذا دخل مستخدم (أ) و قام بكتابة كلمة  
و دخل مستخدم (ب) فسوف يجد الكلمة التي كتبها المستخدم (أ) - و طبعا هذا غير صحيح بالنسبة للحفظ على خصوصية  
المستخدمين- ؛ لذلك سوف نستخدم (User)  
٢- الثاني (User) هو مخصص للمستخدم بحد ذاته .  
- و لكن كيف سيفرق التطبيق بين المستخدمين !! .. - هذا ما سوف يتم شرحه قريبا في الفصل الخاص بالأمان و  
المستخدمين - .

لنعد مرة أخرى الى موضوعنا؛ بعد هذا المثال الطويل سوف نبدأ رحلة جديدة مع أدوات جديدة و طبعا كلها ذات علاقة بالـ  
(WebParts)

و بعض هذه الأدوات التي سوف تعطى للمستخدم إمكانيات جديدة و بأقل جهد ممكن ؛ هي مالاتى :-

## 1- (CatalogZone)

هذه الأداة تحتوى بعض الأدوات (PageCatalogPart / DeclarativeCatalogPart) ، و هي تقوم بعمل قائمة بالأدوات المتاحة و إتاحة إمكانيات الأدوات إلى هذه الأدوات  
مثال :- يمكن للمستخدم إظهار و إخفاء ما يري من أجزاء الويب بالإضافة لإمكانية إستيراد خصائص الأدوات

وسوف يتم إيضاح هذا التعريف أكثر بمثال إيضاحي

## 2- (DeclarativeCatalogPart)

تتيح للمطور عرض قائمة بالأدوات التي يمكن التعامل معها - إضافة و حذف في أجزاء الويب (WebPart zone)-

## 3- (PageCatalogPart)

إعادة عرض صفحة الأدوات التي تعامل معها المستخدم و قام بإغلاقها مما يتيح له إضافة ما يريد من أدوات جديدة

## 4- (ImportCatalogPart)

هذه الأداة تقوم بإستيراد خصائص الأداة عن طريق الملف الذي قمنا بحفظه من قبل.

Close Catalog Zone

Declarative Catalog

**DeclarativeCatalogPart - DeclarativeCatalogPart1**

Add web parts to this catalog part by editing its WebPartsTemplate by selecting Edit Templates in the context menu

Page Catalog

WebPart 1 ☐

WebPart 2 ☐

WebPart 3 ☐

Imported Web Part Catalog

Type a file name (.WebPart) or click "Browse" to locate a Web Part

Browse...

Once you have selected a Web Part file to import, click the Upload button

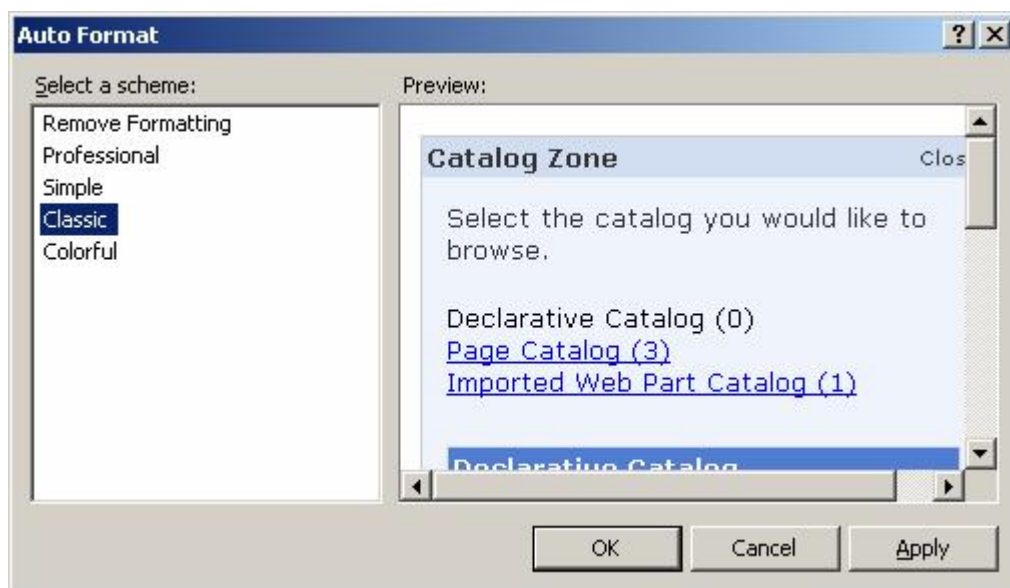
Upload

Imported Web Part

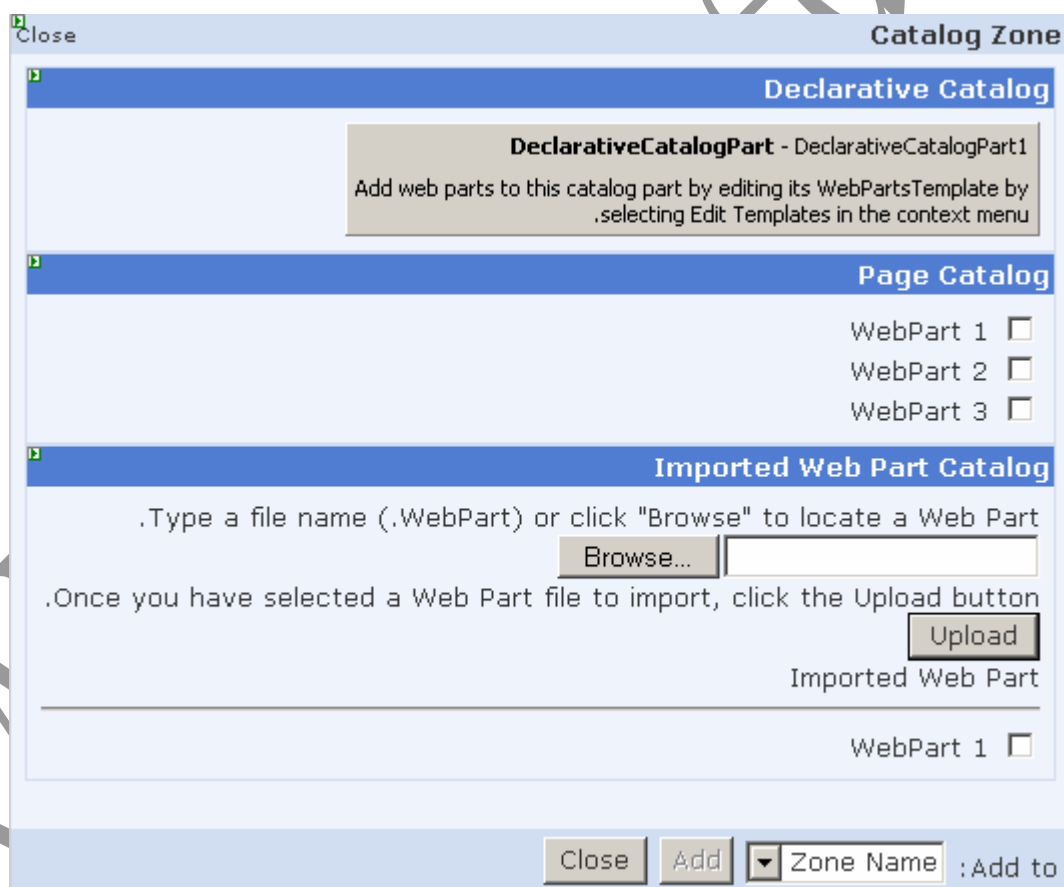
WebPart 1 ☐

Close Add Zone Name :Add to

الشكل 43 [ كما نرى أن الأداة (CatalogZone) تحتوي على بقية الأدوات - إذا حاولت وضع هذه الأدوات خارج (CatalogZone) لن تعمل هذه الأداة ]



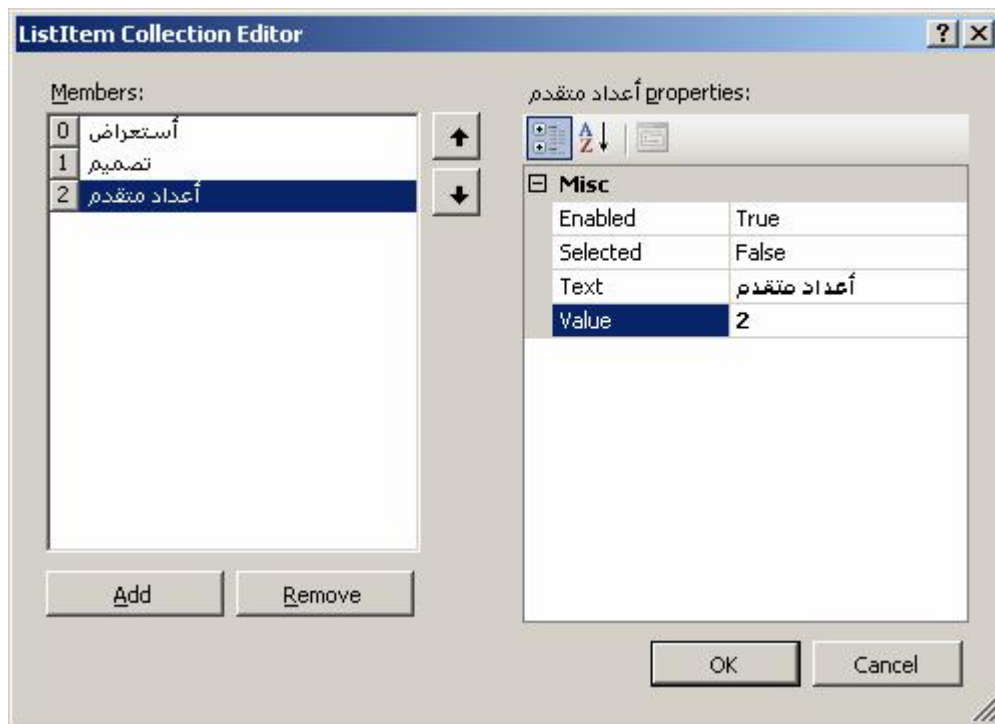
الشكل 44] فقط للتذكير أن تغيير الشكل قد يكون مهم و مريح نفسياً ]



الشكل 45] يوضح الشكل بعد تغيير Style

و الآن لنرى كيف تعمل هذه الأدوات سوياً

أول خطوة نقوم بإضافة عنصر جديد للقائمة المنزلة ( Drop down list ) كما نرى



الشكل ٤٦

```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    switch (int.Parse(ddl_ShowType.SelectedValue))
    {
        case 0:
            WebPartManager1.DisplayMode = WebPartManager.BrowseDisplayMode;
            break;
        case 1:
            WebPartManager1.DisplayMode = WebPartManager.DesignDisplayMode;
            break;
        case 2:
            WebPartManager1.DisplayMode = WebPartManager.CatalogDisplayMode;
            break;
        default:
            WebPartManager1.DisplayMode = WebPartManager.BrowseDisplayMode;
            break;
    }
}
```

#### الكود بصيغة C#

```
Protected Sub DropDownList1_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles ddl_ShowType.SelectedIndexChanged
    Select Case Val(ddl_ShowType.SelectedValue)
```

Case 0

WebPartManager1.DisplayMode = WebPartManager.BrowseDisplayMode

Case 1

WebPartManager1.DisplayMode = WebPartManager.DesignDisplayMode

Case 2

WebPartManager1.DisplayMode = WebPartManager.CatalogDisplayMode

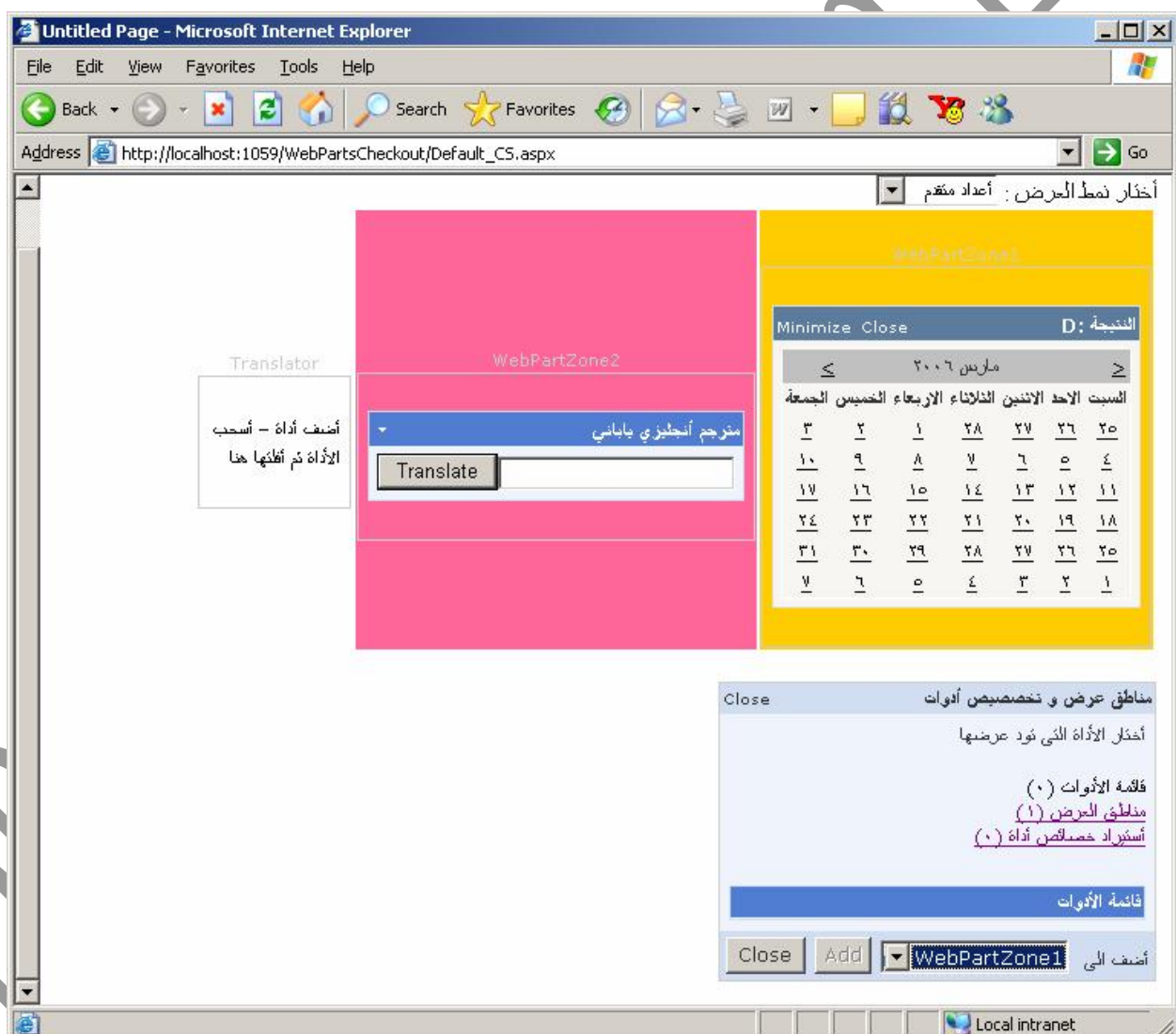
Case Else

WebPartManager1.DisplayMode = WebPartManager.BrowseDisplayMode

End Select

End Sub

### الكود بصيغة VB.net

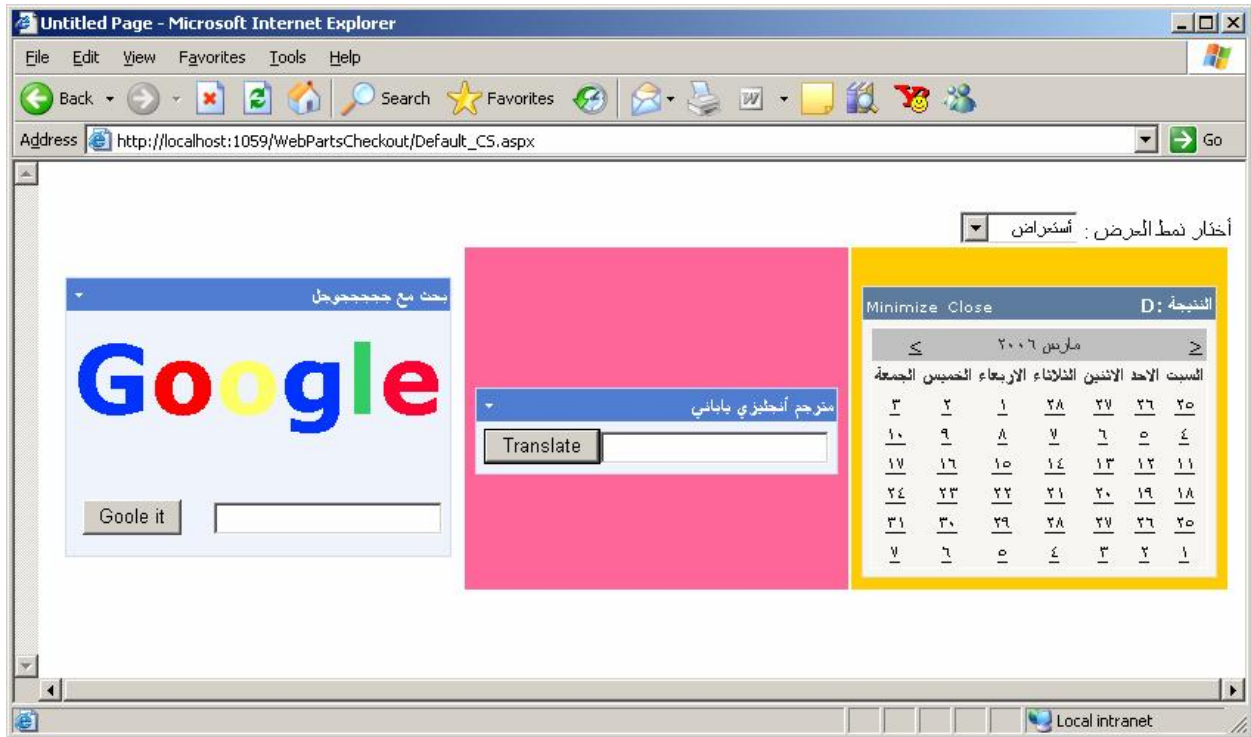


الشكل 47 [ هذا هو الشكل عند تنفيذ التطبيق كما نلاحظ أنه لا يوجد أدوات في قائمة الأدوات و في مناطق العرض يوجد أداة ]



[معرض]





الشكل 49 بعد ذلك بمجرد العودة إلى نمط الاستعراض يتم حفظ كل شيء في قاعدة بيانات الـ (Personalization)



الشكل 50 [تحكم كامل في محتويات الصفحة و بأقل كمية كود ]

هناك ما هو أقوى من ذلك وهو أن نعطي المستخدم القدرة على تعديل الأدوات الموجودة في الصفحة ؛ كأنه يعمل على الـ ( VS ) ، بمعنى أن المستخدم سوف يكون قادر على تغيير كل خصائص العناصر الموجودة في الصفحة كالعادة - يبدو الكلام غير واضح لـ - و لكن لاداعي للقلق لأنه هناك مثال لتوضيح هذه النقطة و الآن لنكمل المثال.

- سوف نضيف بعض الأدوات للمثال

#### 1- (EditorZone)

هذه الأداة هي التي سوف تحتوى على كل أدوات التعديل و سوف تكون بمثابة ( Interface ) بين الـ ( WebPartManager ) و أدوات التعديل .

## 2- (PropertyGridEditorPart)

سوف تعطي هذه الأداة القدرة على تعديل الخصائص المضافة إلى الأداة ( Custom Property ) التي يقوم بكتابتها المطور.

## 3- (LayoutEditorPart)

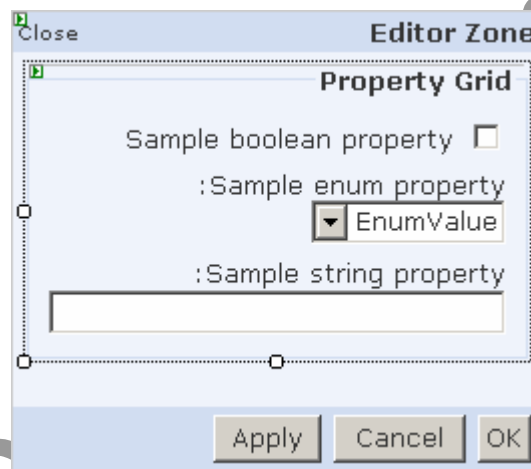
وهي تعطي المستخدم القدرة على تحديد مكان عرض الأداة بالإضافة إلى بعض خصائص العرض.

## 4- (AppearanceEditorPart)

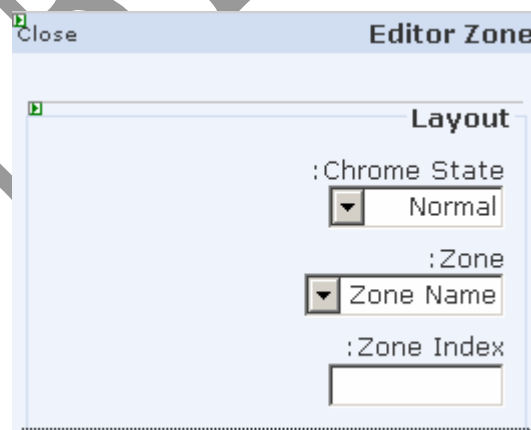
لهذه الأداة القدرة على تعديل الأداة بشكل رائع مع القدرة على تحديد التفاصيل بدقة

## 5- (BehaviorEditorPart)

وهي تعطي السيطرة على خصائص بشكل أكثر دقة ، و سوف يتضح في المثال



الشكل ٥١



الشكل ٥٢

Close

Editor Zone

Behavior

:Description

:Title Link

:Title Icon Image Link

:Catalog Icon Image Link

:Help Link

:Help Mode

Modal

:Import Error Message

:Export Mode

Do not allow

:Authorization Filter

Allow Close

☐

Allow Connect

☐

Allow Edit

☐

Allow Hide

☐

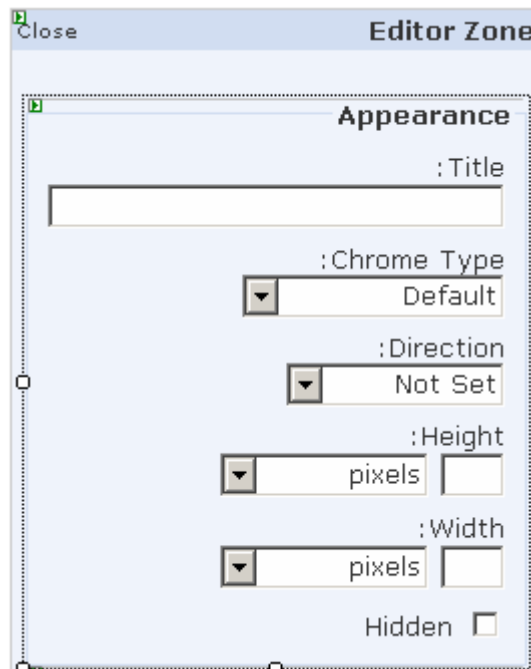
Allow Minimize

☐

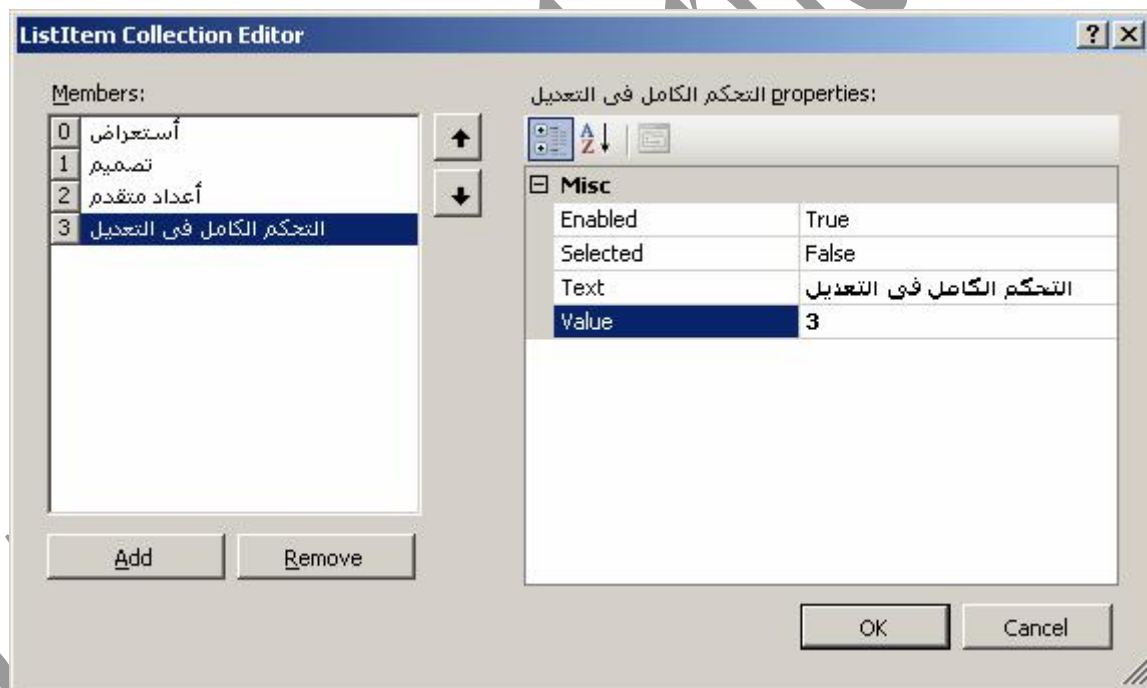
Allow Zone Change

☐

الشكل ٥٣



الشكل ٥٤



الشكل ٥٥ [ نضيف عنصر جديد للقائمة المنزلة ]

```
protected void DropDownList1_SelectedIndexChanged(object sender, EventArgs e)
{
    switch (int.Parse(ddl_ShowType.SelectedValue))
    {
        case 0:
            WebPartManager1.DisplayMode = WebPartManager.BrowseDisplayMode;
    }
}
```

```
        break;

    case 1:
        WebPartManager1.DisplayMode = WebPartManager.DesignDisplayMode;
        break;

    case 2:
        WebPartManager1.DisplayMode = WebPartManager.CatalogDisplayMode;
        break;

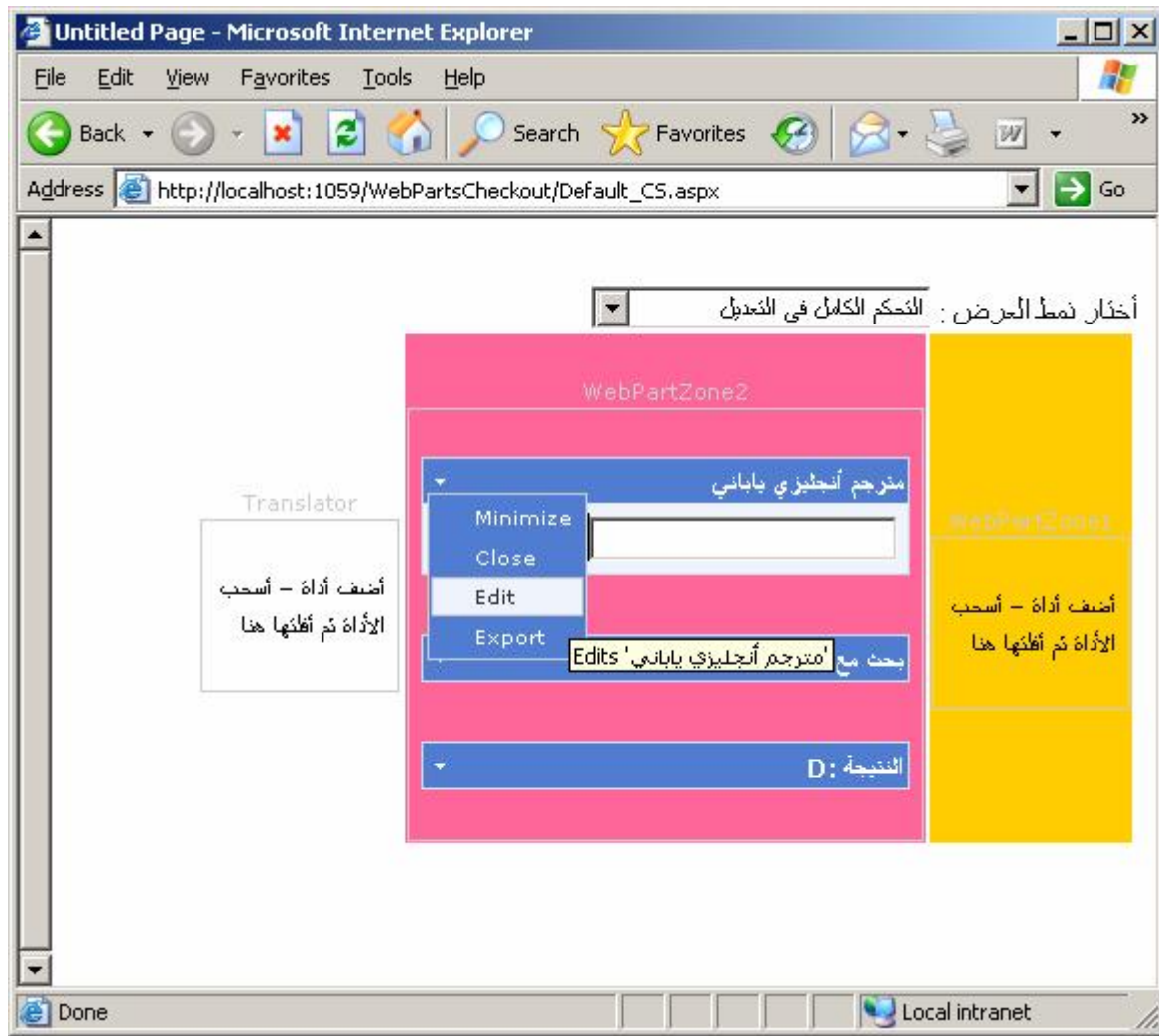
    case 3:
        WebPartManager1.DisplayMode = WebPartManager.EditDisplayMode;
        break;

    default:
        WebPartManager1.DisplayMode = WebPartManager.BrowseDisplayMode;
        break;
}
}
```

الكود بصيغة الـ C#

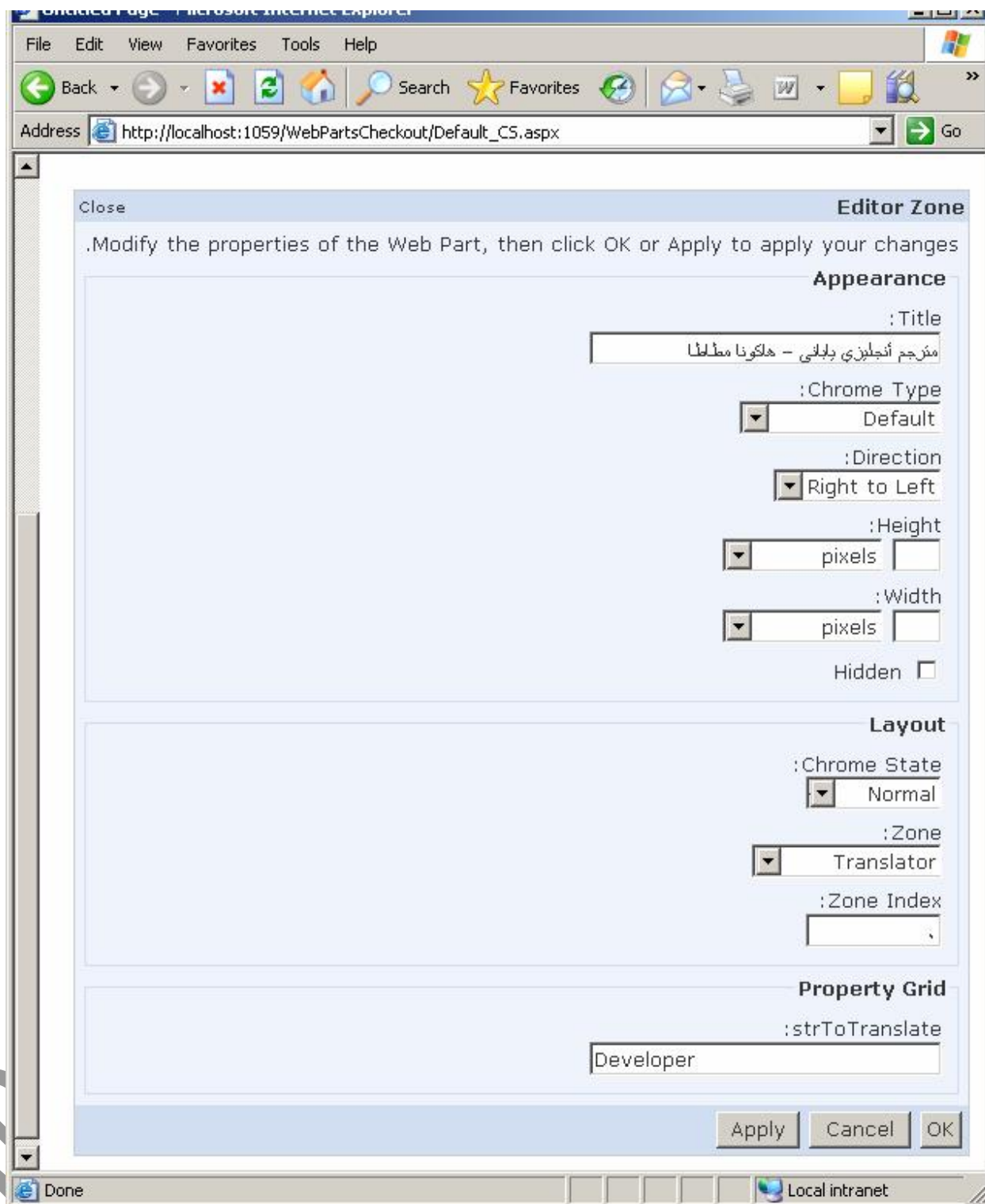
```
Protected Sub DropDownList1_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles ddl_ShowType.SelectedIndexChanged
    Select Case Val(ddl_ShowType.SelectedValue)
        Case 0
            WebPartManager1.DisplayMode = WebPartManager.BrowseDisplayMode
        Case 1
            WebPartManager1.DisplayMode = WebPartManager.DesignDisplayMode
        Case 2
            WebPartManager1.DisplayMode = WebPartManager.CatalogDisplayMode
        Case 3
            WebPartManager1.DisplayMode = WebPartManager.EditDisplayMode
        Case Else
            WebPartManager1.DisplayMode = WebPartManager.BrowseDisplayMode
    End Select
End Sub
```

الكود بصيغة الـ VB



الشكل ٥٦ [ للتعامل مع إمكانية التعديل لابد من اختيار ( Edit ) ، من قائمة كل أداة سوف تظهر القائمة ( Edit ) لكل أداة بعد تغيير نمط العرض الى "التحكم الكامل في التعديل" ]





الشكل 57 [ وفيه تلاحظ أنه يمكن تغيير أي خاصية لأي أداة كل ما عليك بعد التعديل أضغط (OK) ]

ملاحظة :-

هي أنه يمكنك ترجمة كل الكلمات الموجودة في الأدوات - كل هذا - من نافذة الخصائص

(Properties window)



الشكل 58 [ هذا هو الشكل النهائي.]

**و في النهاية هذا الفصل أود أن نتذكر سوياً ملخصاً لما تعلمناه في هذا الفصل:**

١. التعامل مع أدوات العرض و التنسيق الـ (WebParts)
٢. إنشاء أداة من نوع أجزاء ويب (Creating web library control that inherits from WebPart)
٣. التعامل مع أدوات تحريك الـ (WebParts)
٤. التعامل مع أدوات التعديل و تغيير خصائص الـ (WebPart)