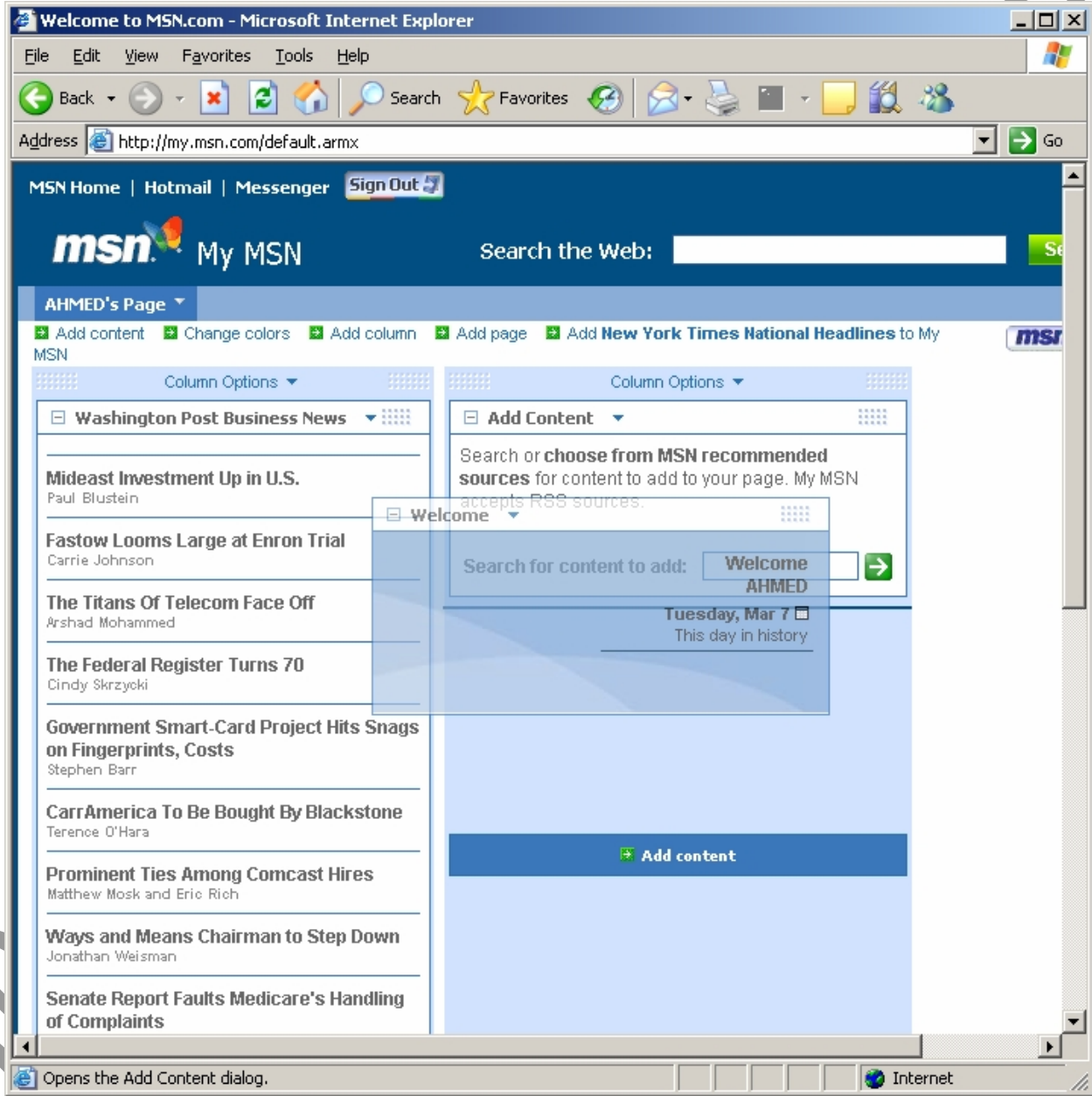


الفصل الثانى : أجزاء الويب و التفاعليه العاليه مع المستخدم.

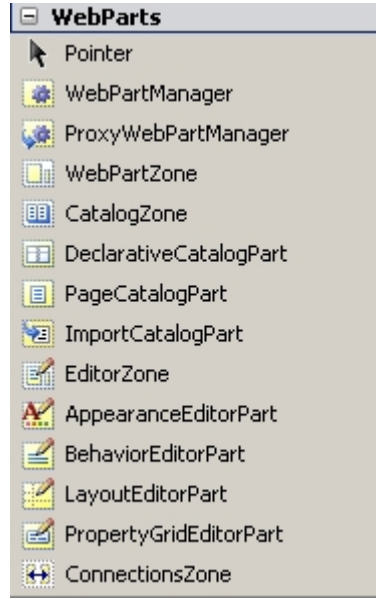
Web Parts, and rich user interface

فى 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 تلاحظ فى الشكل التالى أنه يمكنك نقل المكونات من مكان الى آخر بالإضافة الى إمكانية تغيير الألوان.



الشكل 1

فى الشكل التالى موضوع مجموعة الأدوات الجديدة الخاصة بهذا الموضوع، هذه الأدوات لها طريقة معينة للتعامل ليس مجرد سحب و أفلات للعناصر على الصفحة و لكن الموضوع، ليس بسيط و ليس معقد يمكننا أن نقول "السهل الممتنع"، لكن لا تقلق فالموضوع إذا تم أخذه بتسلسلة الطبيعي سوف نصل الى نتائج ممتازة جدا أكثر مما كنت تتوقع.



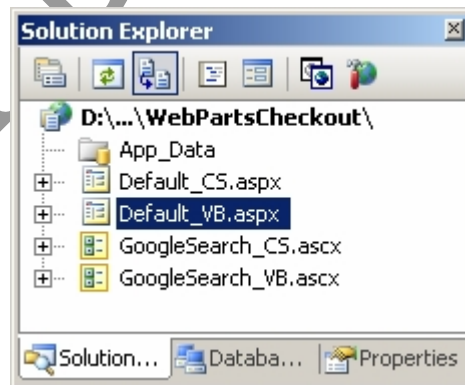
الشكل ٢

و الآن لابد من معرفة بعض النقاط و الخطوات المهم للتعامل مع هذه الأدوات و سوف يتم شرح الأدوات واحدة تلو الأخرى بكل متطلباتها و سوف يكون هناك مثال واضح لكل أداة من الأدوات.

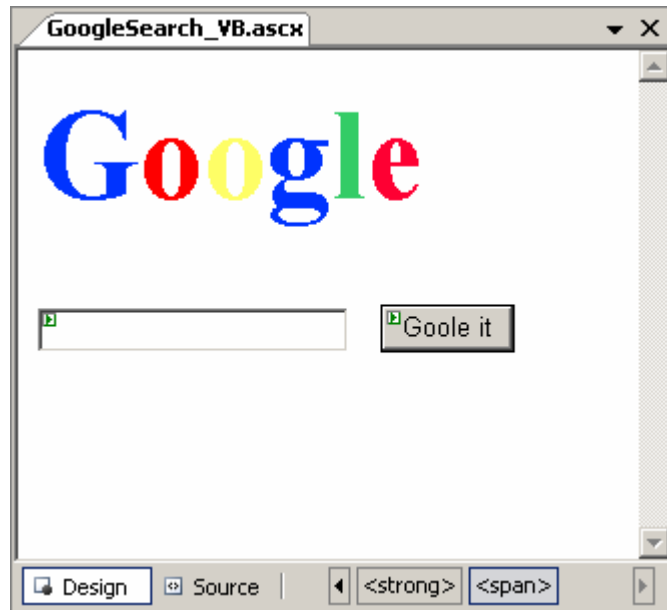
والآن سوف سوف نبدأ بمثال صغير و سوف نستعمل فيه كلا من الأدوات **WebPartManager** و **WebPartZone** و لا بد من توضيح شي بالنسبة لكل من الأداةين :

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

يبدو التعريف السابق الى حد ما و لكن لاتقلق فالمثال خير دليل لتوضيح الأمور و الآن سوف نقوم بتطبيق مثال و سوف نرى ما يمكن عمله مع أجزاء الويب.



الشكل ٣ قم بإضافة الملفات كما هو موضح في الشكل.



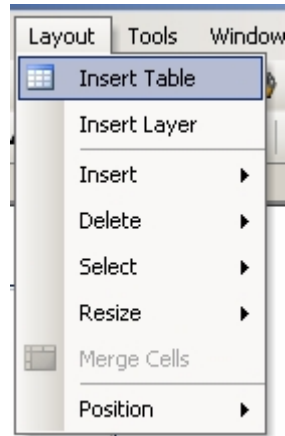
الشكل ٤ : في الـ **User Control** نقوم بعمل الشكل التالي "وظيفة الأداة هو البحث بواسطة جول"

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

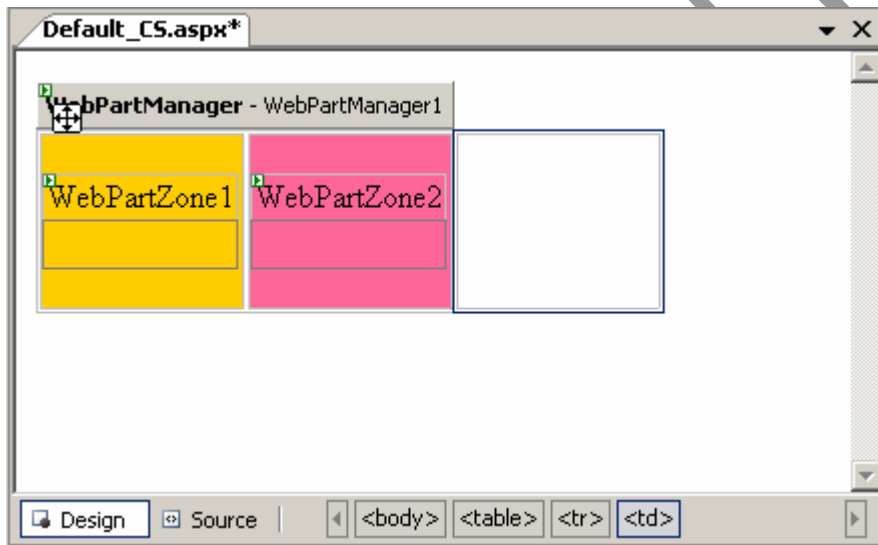
الكود بالـ **C#**

```
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
```

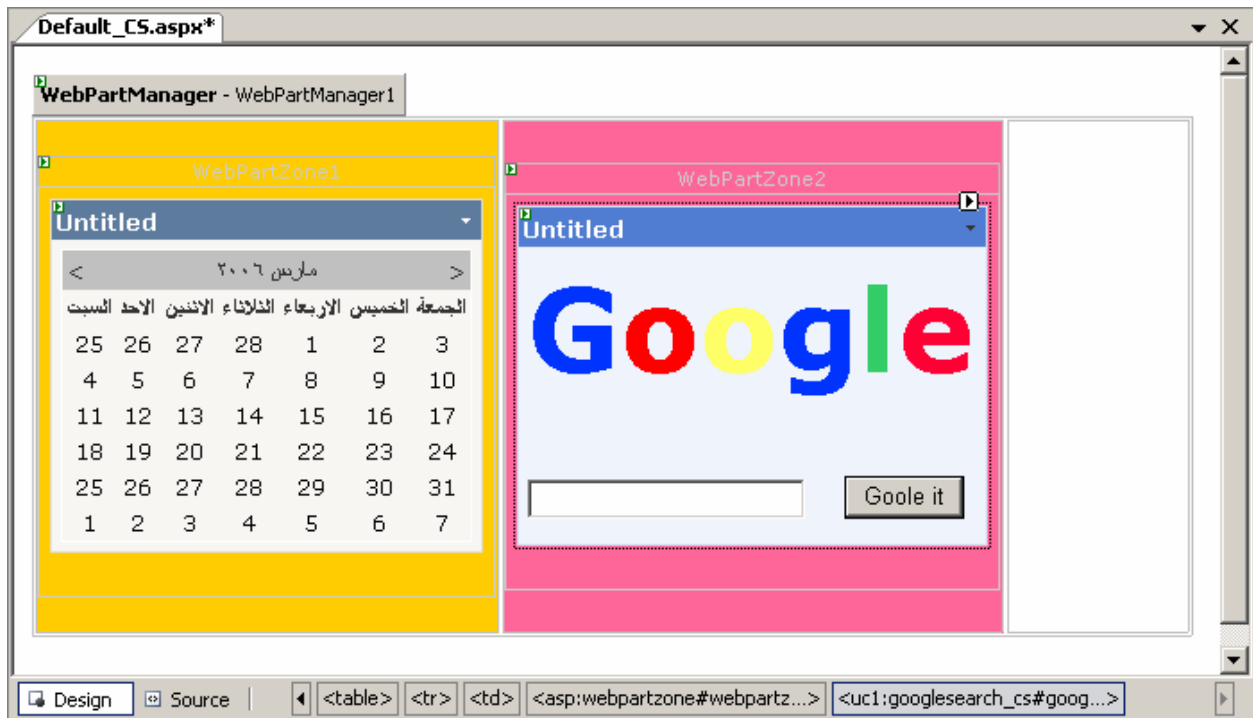
الكود بالـ **VB.net**



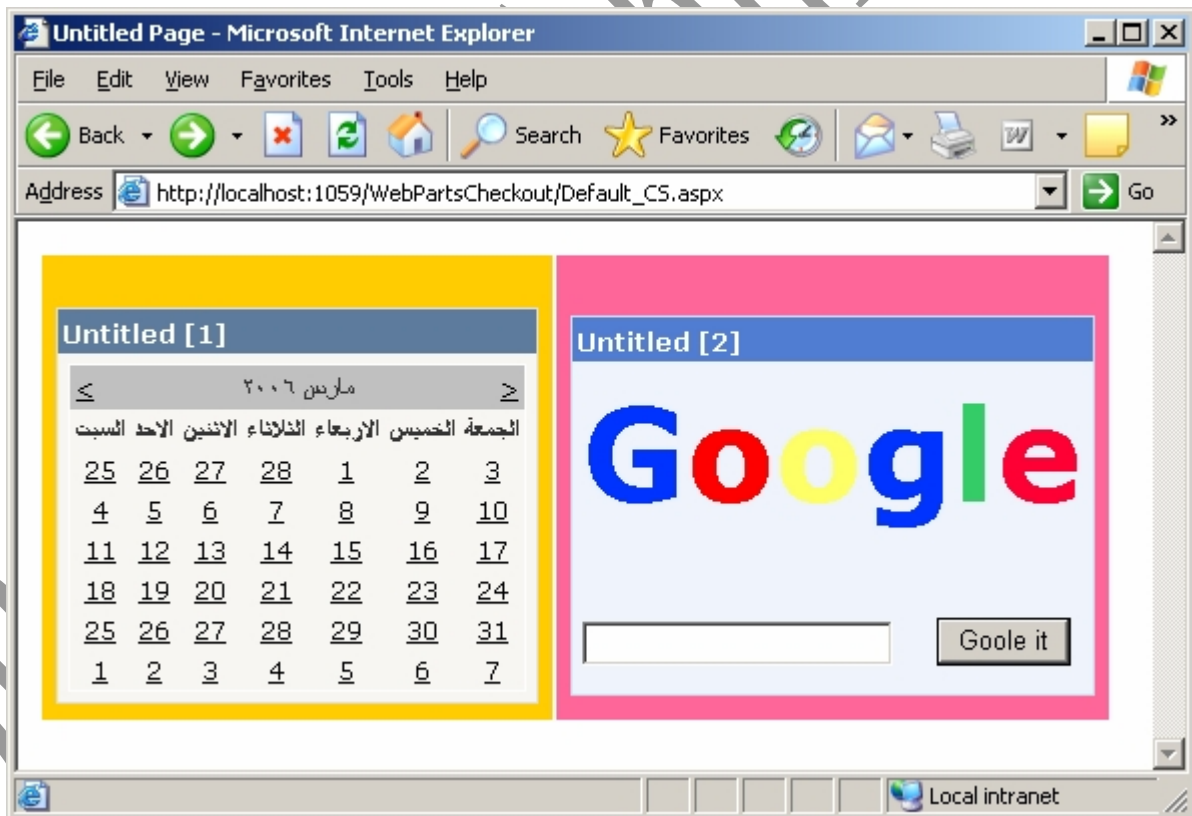
الشكل ٥ نقوم بإضافة جدول صفحة



الشكل 6 و نقوم بإضافة كلا من الأداة WebPartManager و WebPartZone



الشكل 7 نقوم بأداة المبراد اضافتها داخل WebPartZone نقوم بأضافة أدوات النتيجة ثم UserControl.



الشكل 8. هذا هو الشكل عن تشغيل المثال و لكن كما نلاحظ أن العنوان مكتوب Untitled و لتغييره أدخل على الـ source view

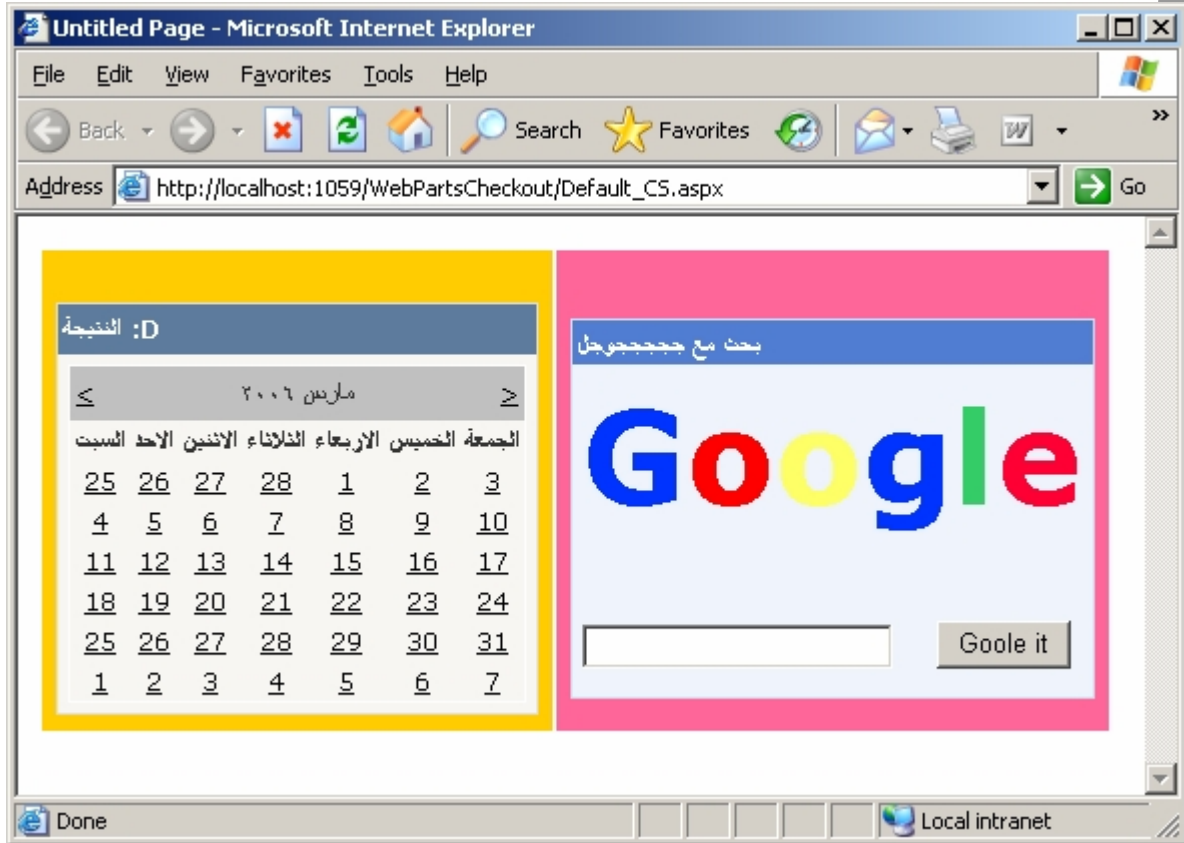
```
<uc1:GoogleSearch_CS ID="GoogleSearch_CS1" title="جوجل مع بحث"
runat="server" />
```

Title="D: النتيجة"

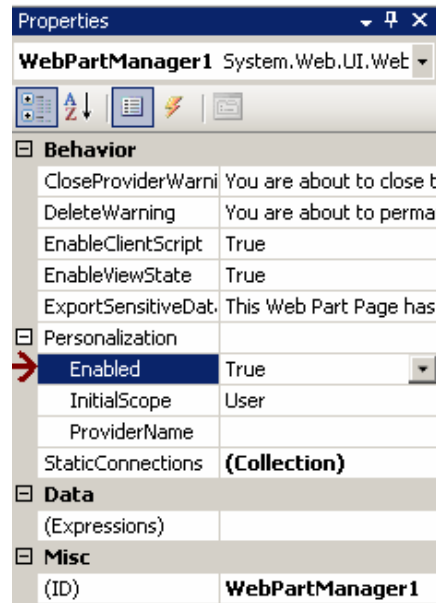
الشكل 9 أضاف التوصيف Title سوف تلاحظ رفض ال VS التعديل الذي قمت به و لكن لا تقلق سوف توضيح كل شيء لاحقاً

title="بحث مع جججججوجل"

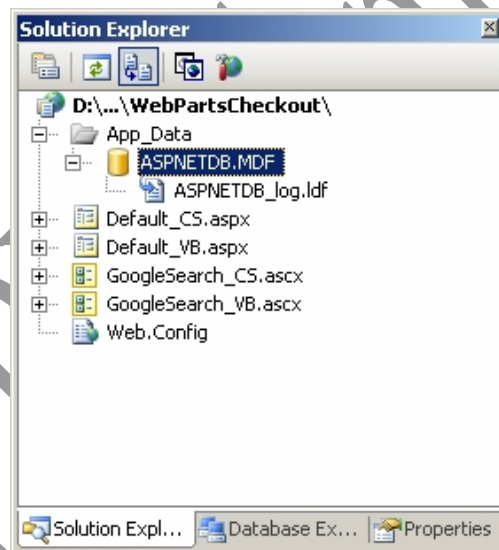
الشكل 10



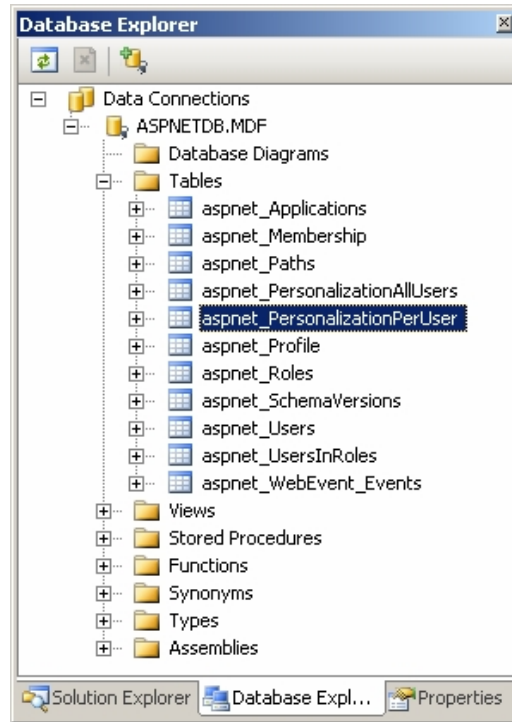
الشكل 11 هذا هو الشكل الجديد بعد تعديل العنوان – و لكن حتى الآن لم نستفيد من مميزات WebParts و الآن سوف نوضح كيفية التعامل مع WebParts و الاستفادة من مميزات الجبارة في ال-**Personalisation**



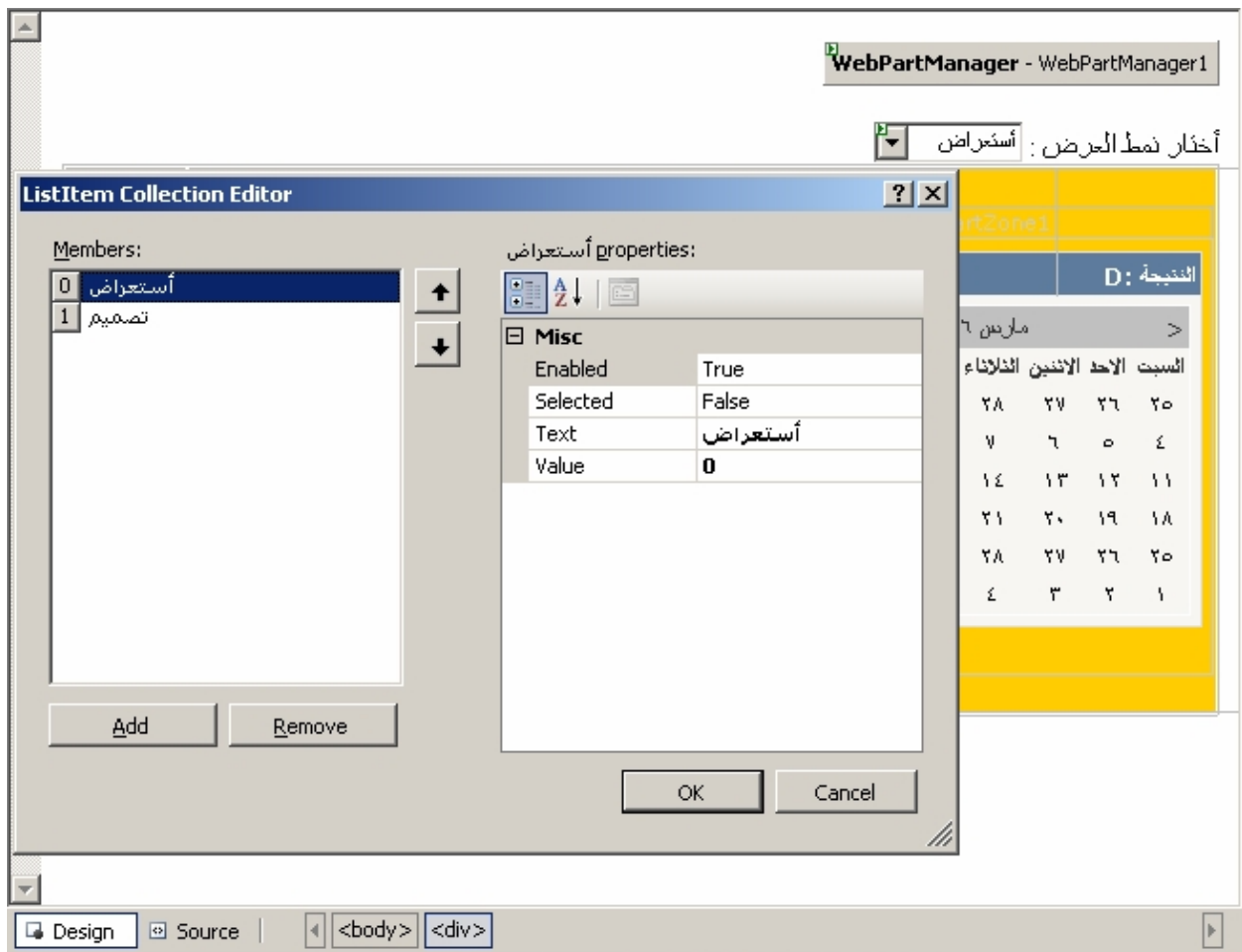
الشكل 12 قم بتنشيط الـ **Personalization** الى **True** – سوف يتطلب تفعيل هذه الخاصية الى وجود قاعدة بيانات الـ **Membership and personalization** كما هو موضح في الشكل التالي.



الشكل ١٣ هذه هي قاعدة البيانات التي تقوم بإدارة كل إمكانيات الـ **Membership** و الـ **Personalization**



الشكل 14 بما أن الـ VS يتكامل مع الـ MS-SQL Server سوف تلاحظ أماكن كثيرة و منها أن معظم أماكن الـ SQL server موجودة في الـ VS.net



الشكل ١٥ قم بإضافة قائمة منزلقة و أضف فيها عنصران كما هو موضح (أستعراض = ٠ / تصميم = ١) و قم بتنشيط **Autopostback** الى **True** للقائمة المنزلقة، ثم أضف الكود التالي.

```
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;
    }
}
```

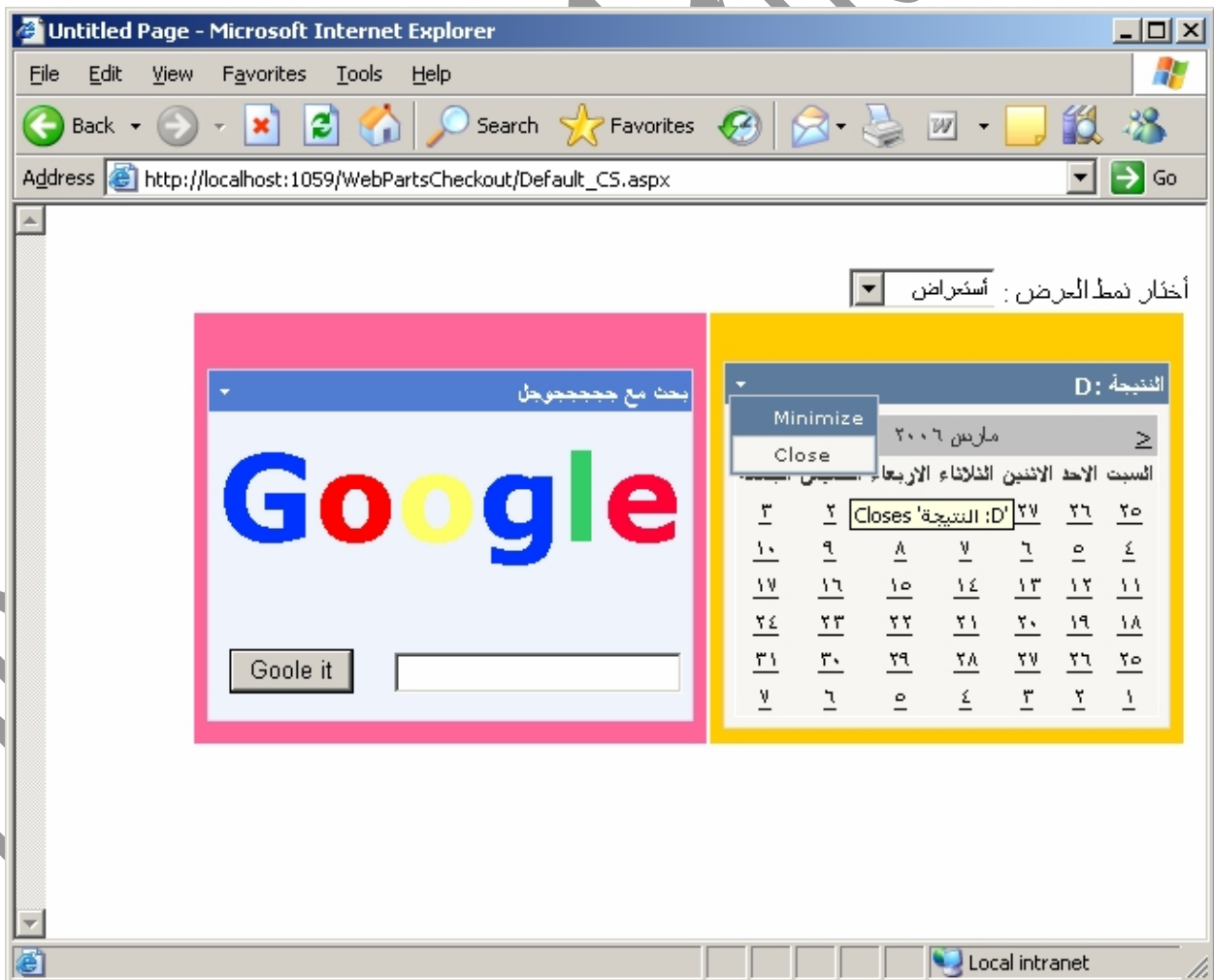
الكود بالـ C#

```

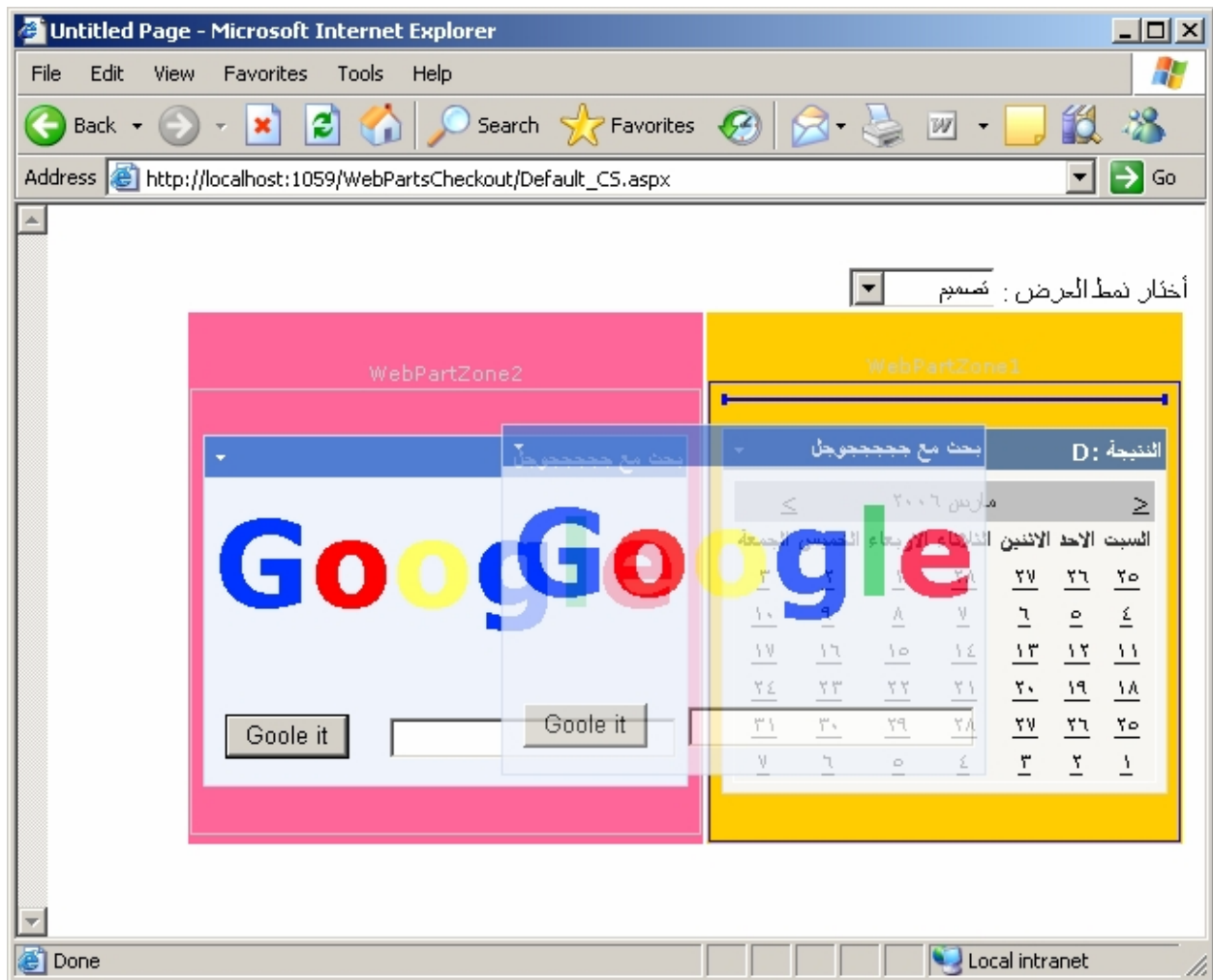
Protected Sub DropDownList1_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles ddl_ShowType.SelectedIndexChanged
    Select Case Val(ddl_ShowType.SelectedValue)
        Case 0
            WebPartManager1.DisplayMode =
WebPartManager.BrowseDisplayMode
        Case 1
   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

```

الكود بالـ VB.net



الشكل 16 ، هذا هو الشكل عند الانتهاء من المثال يمكنك الآن أن تقوم بأغلاق أو تصغير محتويات الـ WebPartZone



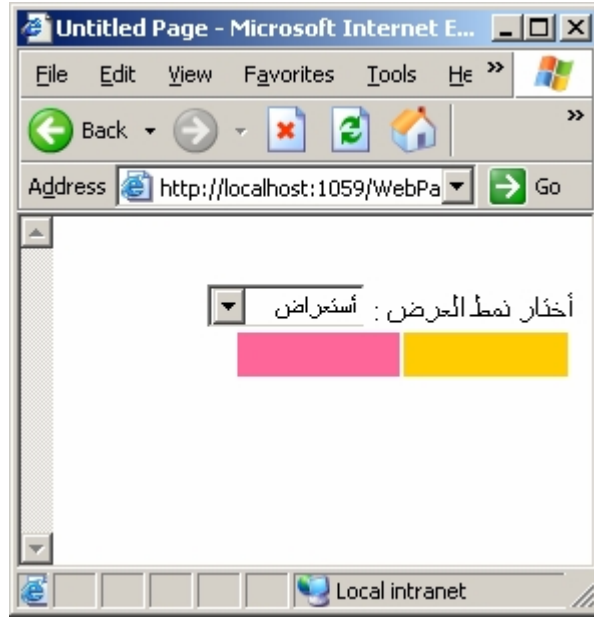
الشكل 17 بالإضافة الى إمكانية تحريك العناصر.



الشكل 18 بكل سهولة يمكنك نقل العناصر من WebPartZone الى الأخرى دون مشاكل.



الشكل 19 بالاضافة الى أن التطبيق يقوم بحفظ أعدادات المستخدم، قم بإغلاق العناصر الموجودة.

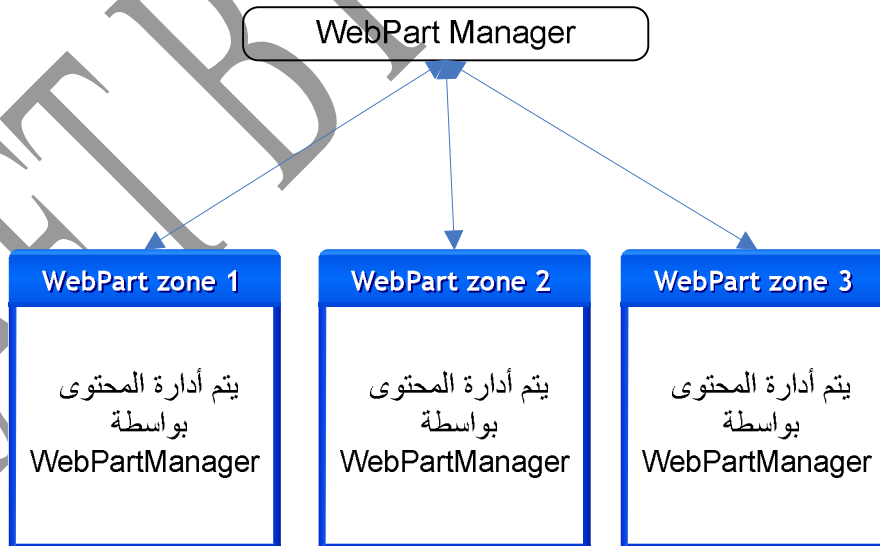


الشكل 20 هناك مشكلة و هي كيف يمكننا أن نظهر المكونات مرة أخرى.

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 ٢١ لا داعي للقلق الموضوع بسيط للغاية هناك طريقتان لحل هذه المشكلة أول وهي الغاء السطر الخاص بالمستخدم من قاعدة البيانات (لا أحبذ هذا الحل) و الحل الآخر هو استعمال مكونات إضافية من الـ **ToolBar** لحل هذه المشكلة.

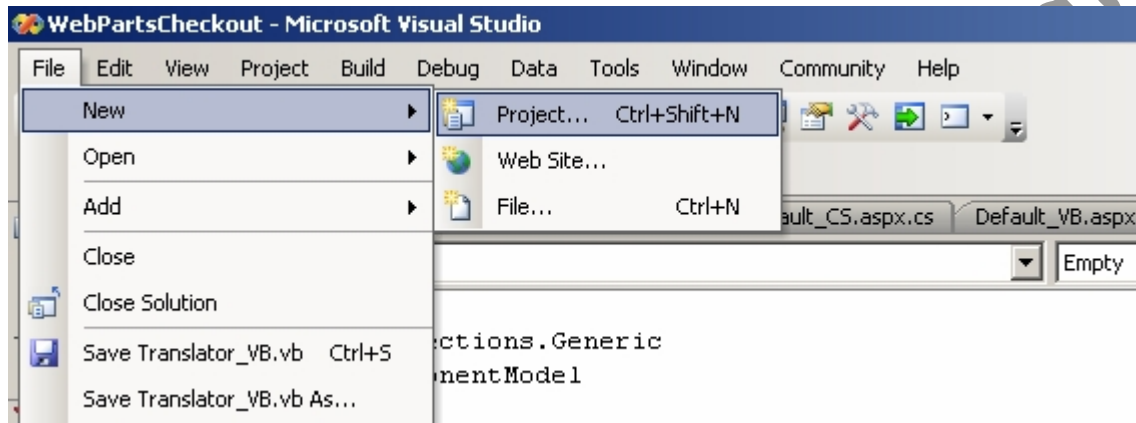
و الآن لنشرح ماذا يحدث بالضبط في هذه العملية ،



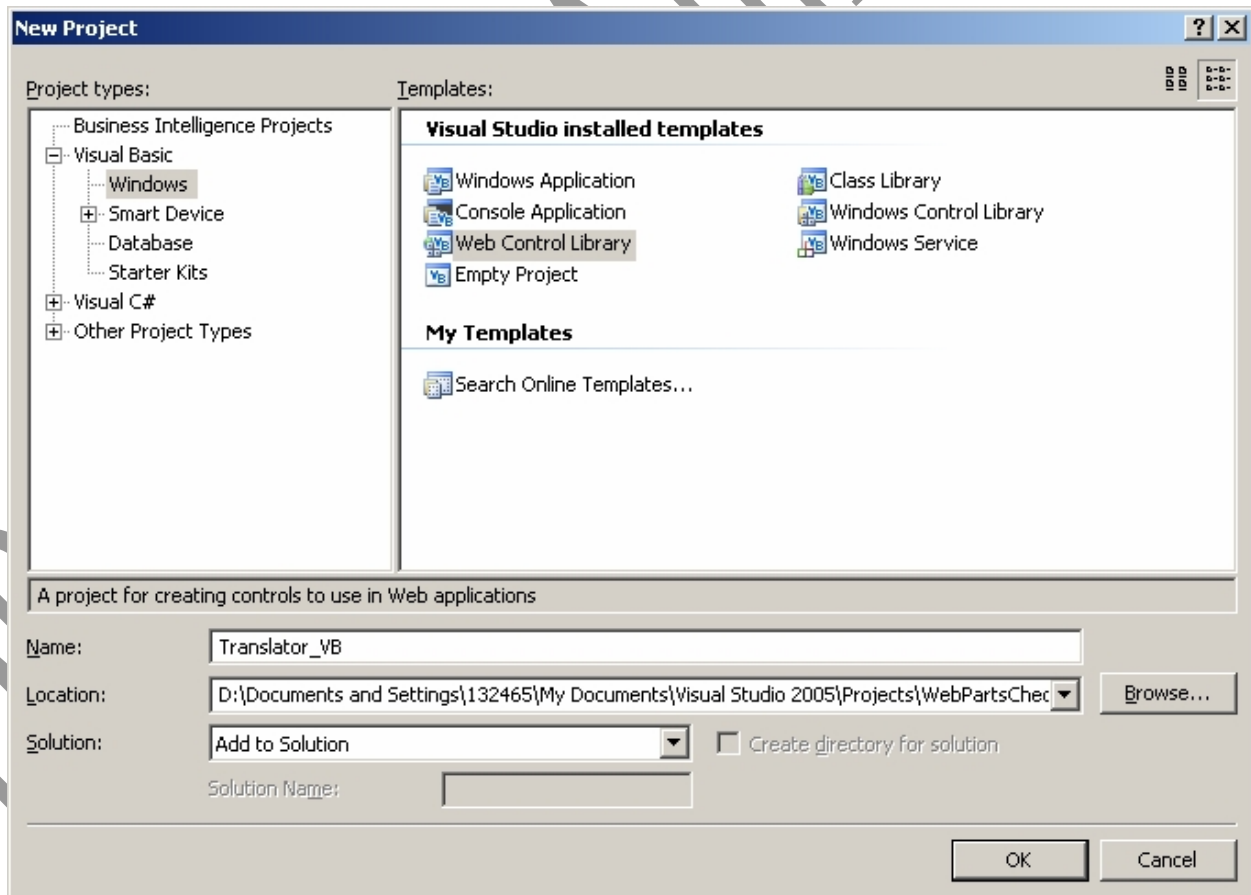
الشكل ٢٢ كما هو موضح في الشكل أن عملية إدارة المحتوى (المحتوى هنا تعني الأدوات الموجودة بداخل الـ WebPart Zone) تتم بواسطة الـ WebPart Manager.

لاحظنا أننا قمنا بأضفة معرف بالاسم Title لتغيير اسم الـ WebPart Zone لابد و أن تعلم شئ بسيط جدا الا وهو ان الـ WebPart Zone تكسب ما بداخلها من أدوات بعض الصفات الجديدة و التي هي فى الأصل موروثه من WebPart .Generic

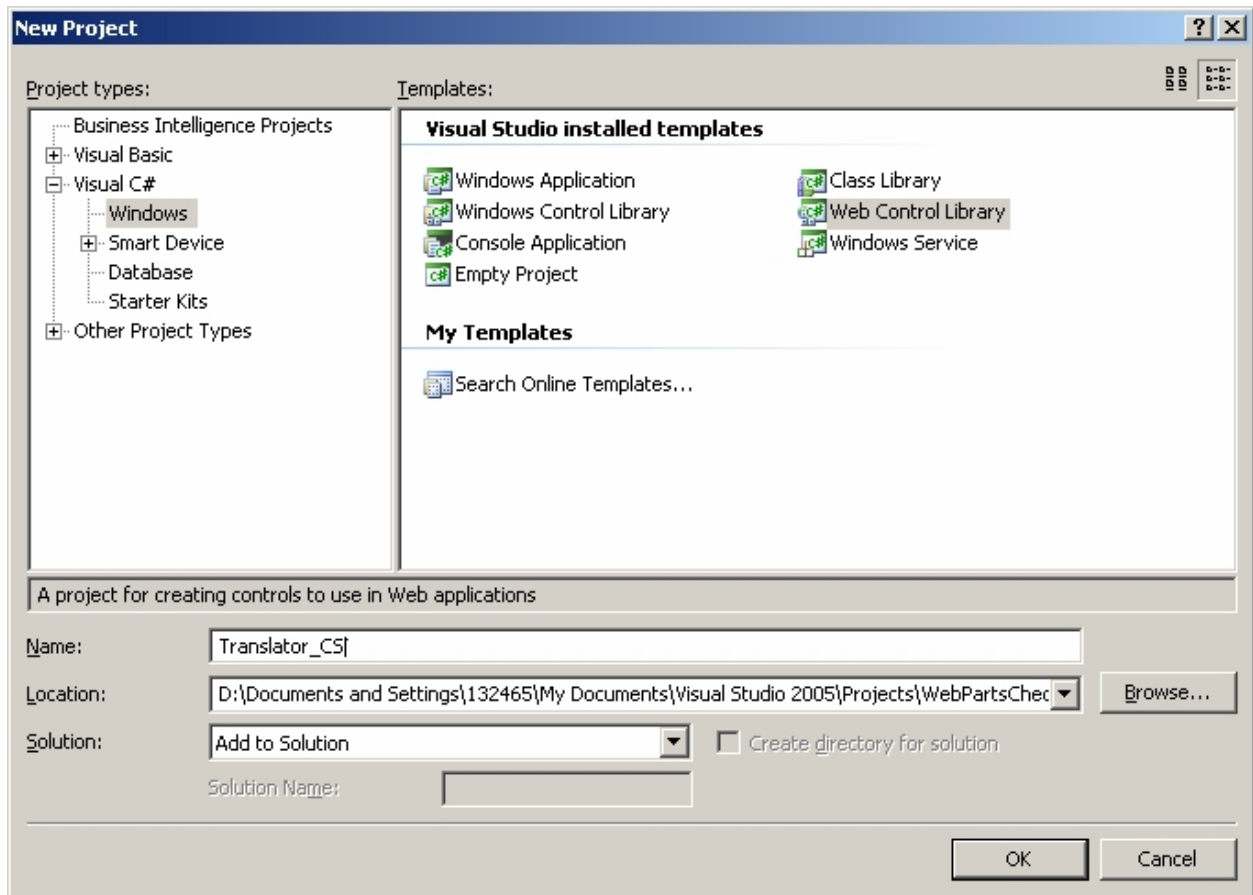
والآن لنبدأ بعمل بعض الإضافات الجديدة على المثال و هي أن نقوم بوراثه صفات المكتبة WebPart لنكسب بعض الإمكانيات التى يمكننا أضافتها لصفحات التطبيق.
و الآن الى المثال و المثال الأتى عبارة عن Custom control يقوم بوراثه المكتبة WebPart و الآن لنبدأ العمل.



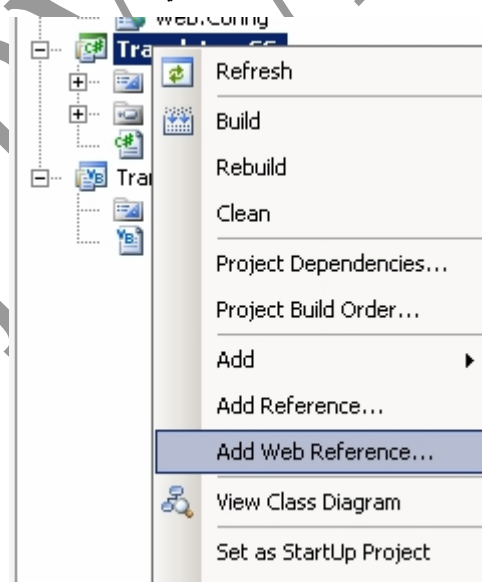
الشكل 23 سوف نقوم بأضافة مشروع جديد من القائمة File à Project .



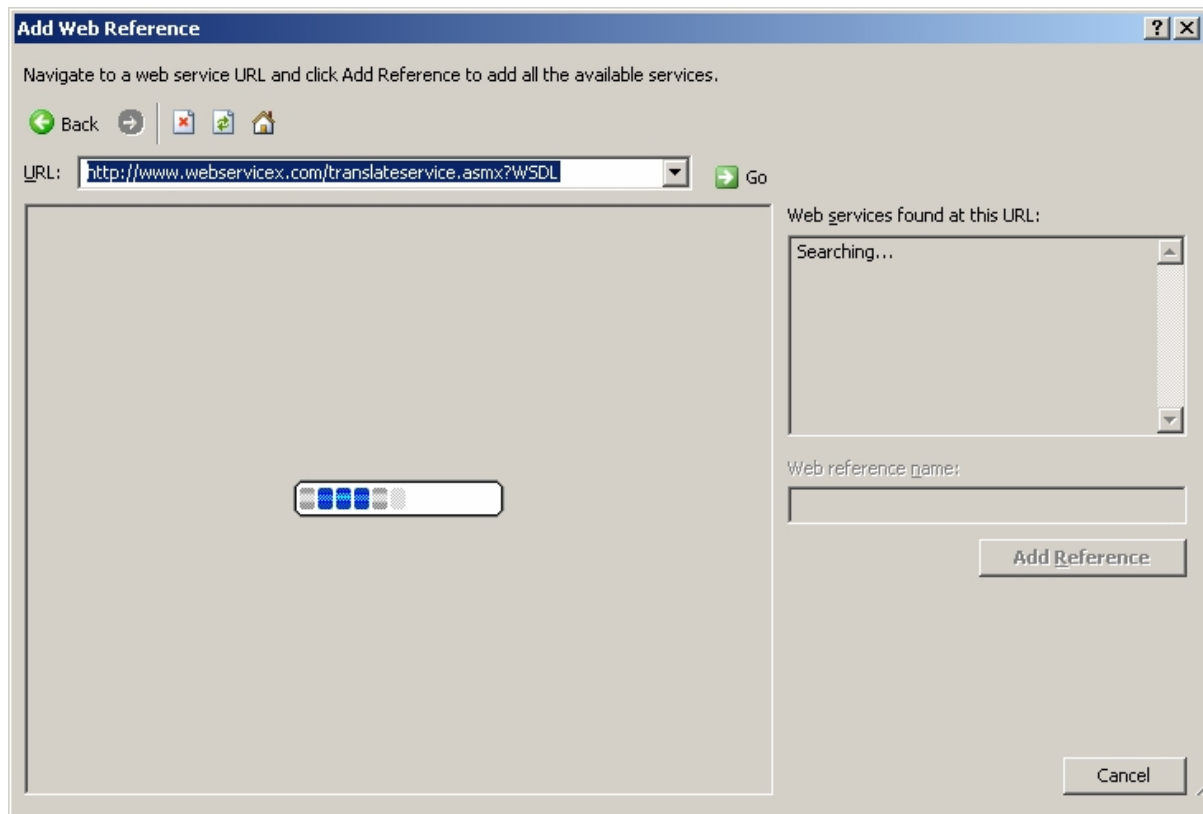
الشكل 24 أختار اللغة مراد التعامل معها هذا الشكل لمبرمجى الـ VB و بعد ذلك نقوم بأختيار نوع المشروع وهو Web Control library



الشكل 25 هذا الشكل لمبرمجي الـ C#

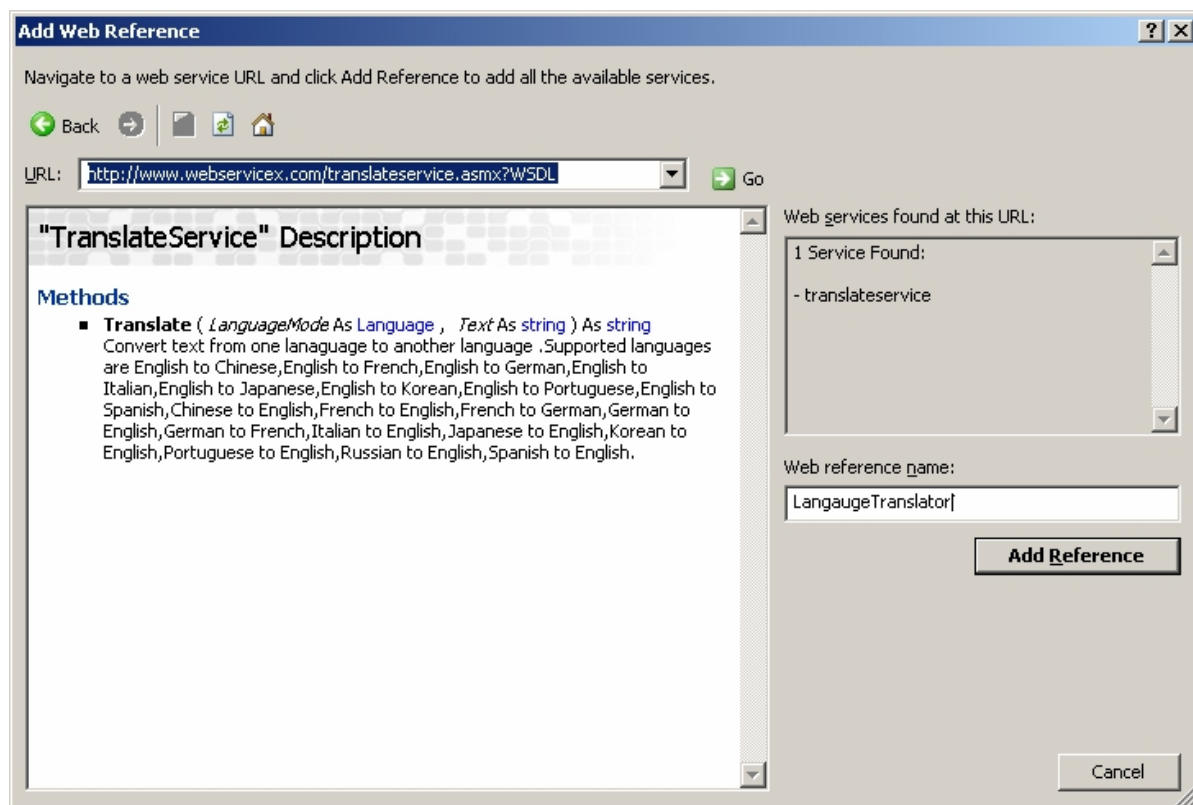


الشكل 26 المثال الذي سوف نطبقه هو عبارة عن مترجم من اللغة الإنجليزية إلى اليابانية باستخدام خدمات الويب **WebService** لهذا سوف نحتاج إلى مرجع ويب **Web Reference** لمزود الخدمة وكالعادة سوف يقوم الـ **Visual studio** بإنشاء المكتبة الوسيطة (**Proxy class**) التي تسهل التعامل مع خدمة الويب **WebService**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

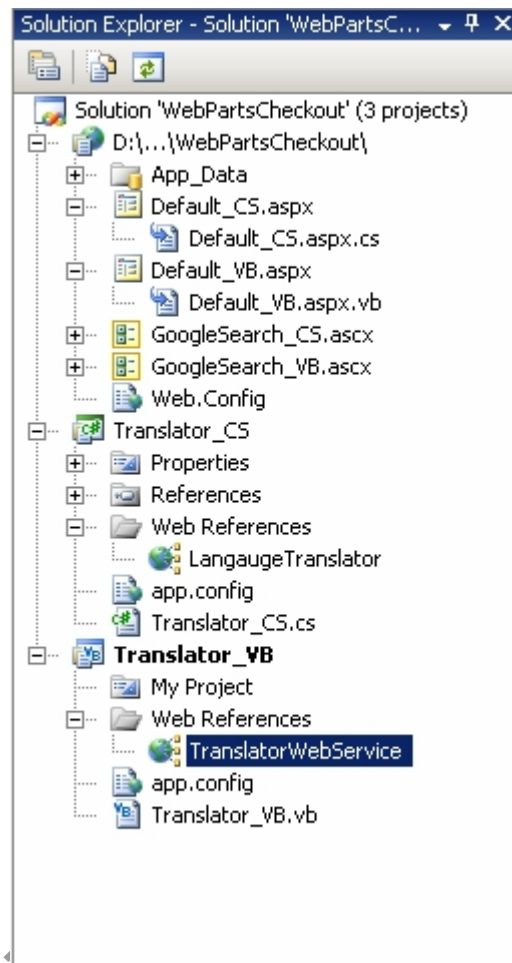


الشكل 27 سوف تظهر هذه النافذة Add Web Reference وسوف نضع عنوان الخدمة المراد التعامل معها و هي

<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 إضافة 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
```

في هذا السطر نقوم بوراثة المكتبة WebPart

متغير من نوع سلسلة حرفيه و التي 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 القابليتين لتخصيص الشخصي Personalization و القابلية للاستعراض WebBrowsable

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

أضافة خاصية للمكتبه و هي الكلمة التي سوف يتم ترجمتها أهمية هذه الخاصية لكي تتيح واجهه برمجية للتعامل مع المكتبه حيث أنه كما 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 الدوات الموجودة مسبقاً

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

أنشاء نسخة جديدة من حقل النص فارغة

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

نسخ محتويات الخاصية strToTranslate الى حقل النص كاقيمة ابتدائية

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

تثبيت حقل النص على الصفحة

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

أنشاء زر جديد و الذي سيتولى الترجمة عن الضغط عليه

```
btnTranslate.Text = "ترجم"
```

تغيير نص الزر الى أسم ملائم

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

أنشاء مناول الحدث Event Handler لكي يقوم بتنفيذ مهمه معينة (التعليمات الموجودة في الدالة btnTranslate_click) و هذا عند الضغط على الزر

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

تثبيت الزر على النافذة

```
lblTranslatedString = New Label()
```

أنشاء أداة نص

```
lblTranslatedString.BackColor = System.Drawing.Color.Yellow
```

تغيير لون الخلفيه الى اللون الأصفر

```
Me.Controls.Add(lblTranslatedString)
```

تثبيت أداة النص على الصفحة

```
ChildControlsCreated = True
```

تأكيد أنشاء الأدوات

```
End Sub
```

```
Private Sub btnTranslate_Click(ByVal sender As Object, _  
ByVal e As EventArgs)
```

أنشاء داله التي سوف تنفذ عند ضغط المستخدم على الزر

```
If txtStringToTranslate.Text <> String.Empty Then
```

التأكد من عدم حلو النص من أى كلمه

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

أحلال قيمة الخاصية strToTranslate بقيمة حقل النص

```
txtStringToTranslate.Text = String.Empty
```

تفريغ حقل النص

```
Dim ws As New TranslatorWebService.TranslateService
```

أنشاء متغير من نوع الخدمة Web Service

```
lblTranslatedString.Text = "<br/>" & Me.strToTranslate & "-->" & _  
ws.Translate(TranslatorWebService.Language.EnglishToJapanese, _  
Me.strToTranslate)
```

تحديد اللغة المراد الترجمة منها و اليها و تحديد الكلمة المراد ترجمتها

```
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 حرفيه 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 للأغلاق

```
    }
```

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

هنا يتم منح و إتاحة الخاصيتين القابليه لتخصيص الشخصي Peronalization و القابلية للاستعراض WebBrowsable

```
    public string strToTranslate
```

أضافة خاصية للمكتبة و هي الكلمة التي سوف يتم ترجمتها أهمية هذه الخاصية لكي تتيح واجهه برمجية للتعامل مع المكتبة حيث أنه كما نلاحظ لا يمكن التعامل مع النص أو حقل النص مباشرة

```
    {
```

```
        get
```

```
        {
```

```
            return pStrText;
```

```
        }
```

```
        set
```

```
        {
```

```
            pStrText = value;
```

```
        }
```

```
    }
```

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

```
    {
```

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

مسح كل الدوات الموجودة مسبقاً

```

txtStringToTranslate = new TextBox();
// إنشاء نسخة جديدة من حقل النص فارغة

txtStringToTranslate.Text = this.strToTranslate;
// نسخ محتويات الخاصية strToTranslate الى حقل النص كقيمة ابتدائية

this.Controls.Add(txtStringToTranslate);
// تثبيت حقل النص على الصفحة

Button btnTranslate = new Button();
// إنشاء زر جديد والذي سيتولى الترجمة عن الضغط عليه

btnTranslate.Text = "Translate";
// تغيير نص الزر الى أسم ملائم

btnTranslate.Click += new EventHandler(btnTranslate_Click);
// إنشاء مناول الحدث Event Handler لكي يقوم بتنفيذ مهمه معينة ( التعليمات الموجودة في الدالة btnTranslate_click ) و هذا 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 الخلفية الى 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 داله التي سوف تنفذ عند ضغط المستخدم على الزر
{
    if (txtStringToTranslate.Text != String.Empty )
        // التأكد من عدم حلو النص من أى كلمه
    {
        this.strToTranslate = txtStringToTranslate.Text;
        // احلال قيمة الخاصية strToTranslate بقيمة حقل النص

        txtStringToTranslate.Text = String.Empty;
        // تفريغ حقل النص

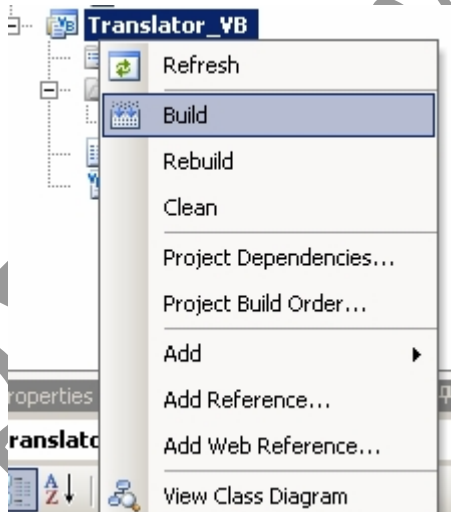
        LanguageTranslator.TranslateService Trans = new
global::Translator_CS.LanguageTranslator.TranslateService();

```

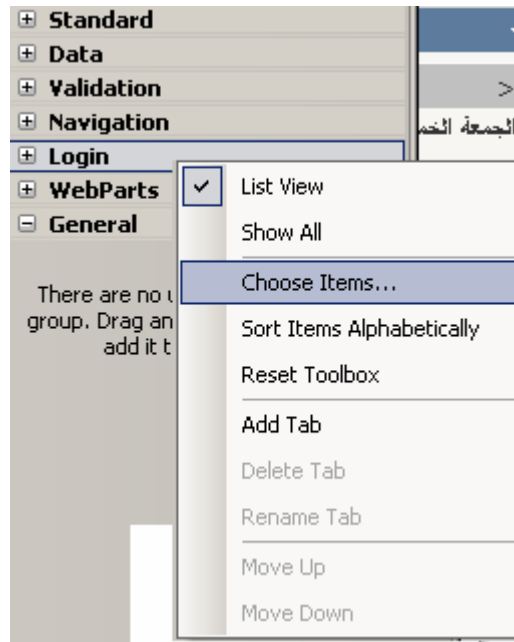
Web Service إنشاء متغير من نوع الخدمة

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

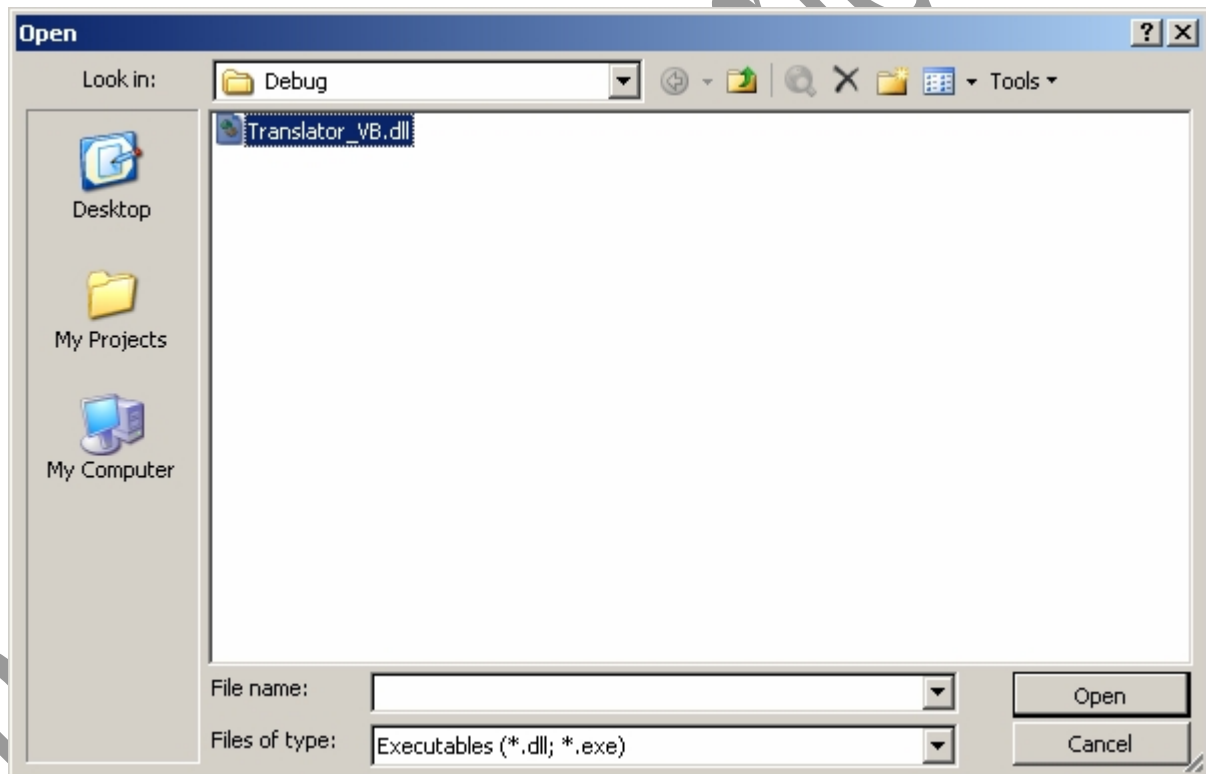
تحديد اللغة المراد الترجمة منها و اليها و تحديد الكلمة المراد ترجمتها



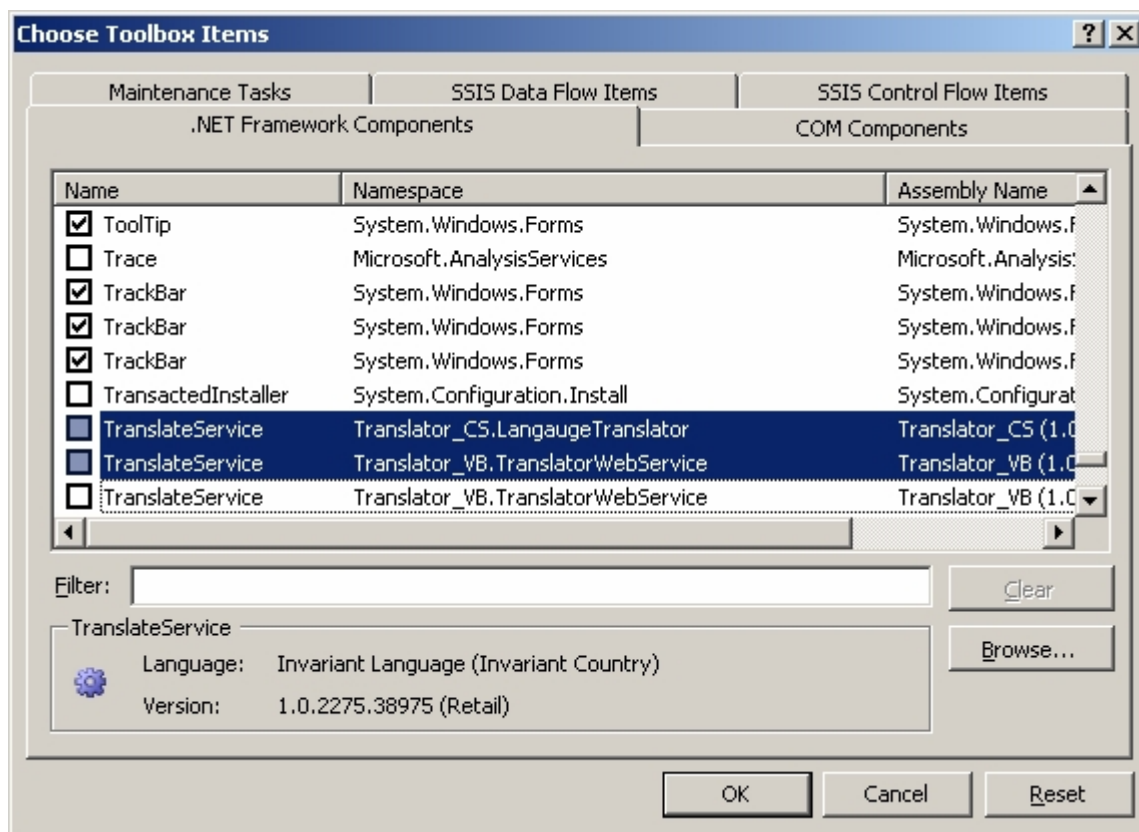
الشكل 30 و الآن و بعد كتابة الكود و الانتهاء من كل المطلوب بالنسبة للأداة نقوم بترجمتها



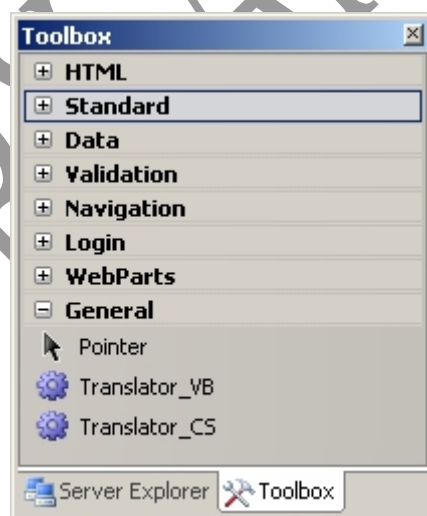
الشكل 31 بعد ذلك نعود للمشروع مرة أخرى و نقوم بأضافة الأداة التي قمنا بتصميمها الآن



الشكل 32 و الآن نختار ملف الذي يحتوي على الأداة و هو موجود في مجلد bin\debug داخل مجلد مشروع المكتبة الذي أنشأناها



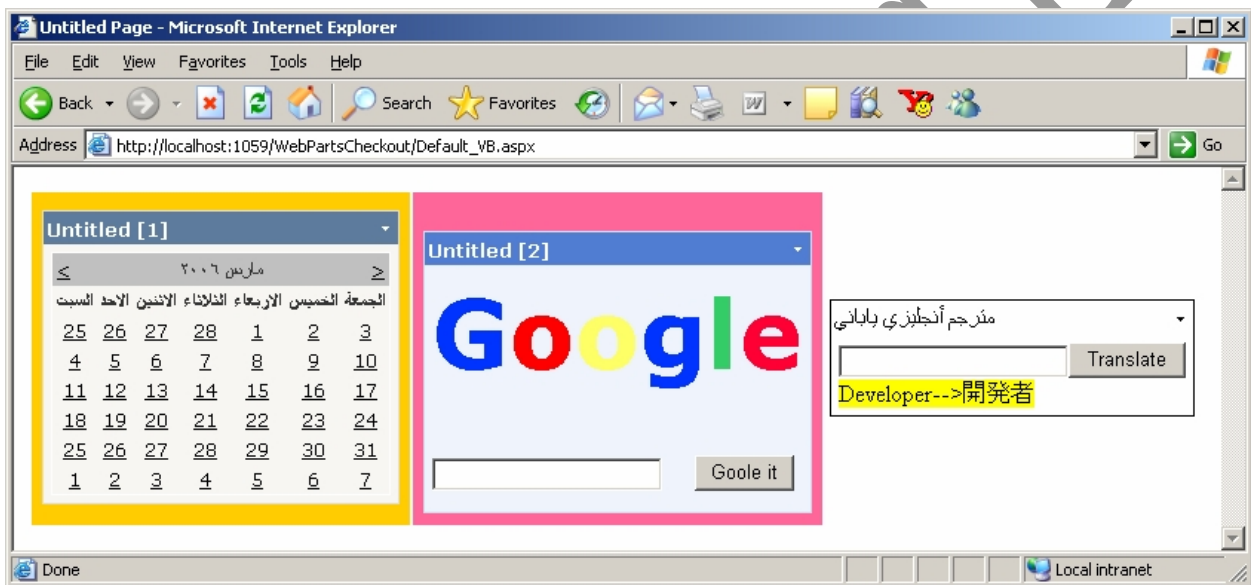
الشكل 33 سوف تظهر كما نرى هنا نقوم بالاختيار ثم ضغط OK



الشكل 34 هذا هو شكل الأدوات بعد إضافتها الى صندوق الأدوات Tool Box

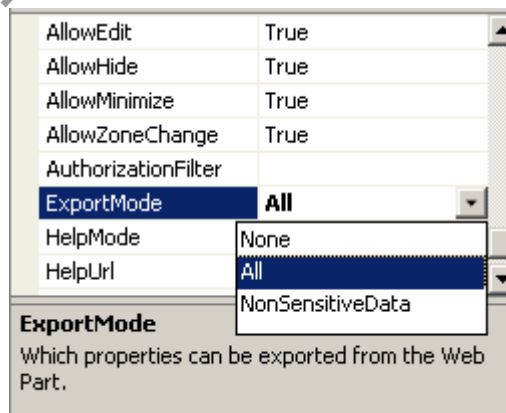


الشكل 35 و الان و بعد أضافت WebZone جديديه تحتوى على الأداة الخاصة بنا هذا هو الشكل فى التصميم



الشكل 36 وهذا هو الشكل عند التنفيذ.

هناك خاصي جديدة و مميزة (من وجهة نظري) وهى أمكانية تمكين المستخدم من أستيراد و تصدير خصائص الأداة و لنرى كيفية تشغيل هذه الأمكانيه كما هو فى الشكل التالى



الشكل 37

و الآن سوف نلاحظ وجود ثلاث خيارات و هم None و هذا معناه عدم اظهار الخاصية عند المستخدم و الثانيه ALL و معناها أستيراد و تصدير كل الخصائص و لكن ماذا لو كانت الأداة تحمل بيانات سرية و خاصة عن المستخدم هنا يأتي الخيار NonSensitiveData ليكن الخير 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

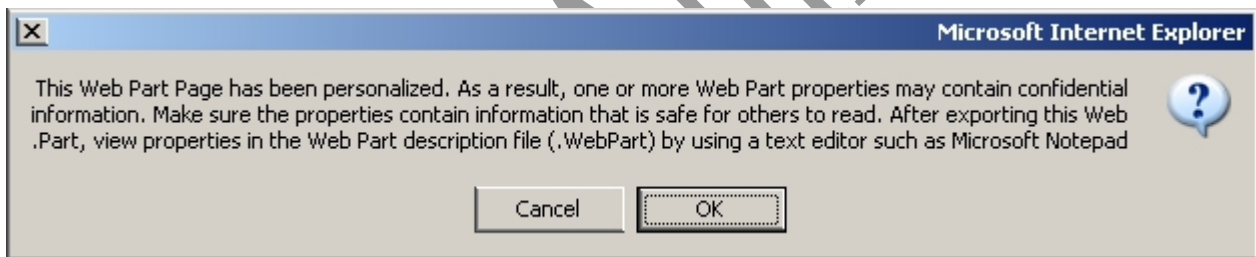
```
<!--
<webParts enableExport="true" />
-->
```

الشكل 38

و الآن لنرى كيف سيتعامل المستخدم مع هذه الخاصية و كيف يستفيد منها



الشكل 39 أول خطوة هي من في 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 الآن نقوم بحفظ ملف الاعدادات 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>
        <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 أستيراد خواص للأداة الخاصة بنا هذا ما سوف نتحدث عنه لاحقا، هناك أيضا أجزاء كثيرة نحتاج الى معرفتها لكي مكسب أمكانيات أكثر.

عند تنفيذ هذا المثال سوف تلاحظون أنه هناك شئ ربما يكون ناقصا الا وهو أنه عند البحث تختفى كلمة البحث ماذا لو أردنا أن تكون هذه الكلمة تظهر مع المستخدم و هنا سوف نعتمد على الـ **Personalization** بشكل قوى جدا و هنا

سوف محتاج الى تعديل كود بسيط في مكتبة الاداة الخاصة بالترجمة لتقوم بتوفيه هذا الغرض وهى أننا سوف نحدد نطاق التخصيص **Personalization Scope** و كما نرى فى الشكل التالى كيف يمكننا ذلك

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



الشكل ٤٢

وصوف نلاحظ وجود نوعان منها و هو مشترك **Shared** و مخصص لمستخدم **User** و الفرق بينهم أن الأول عند استخدامه يتم حفظ آخر كلمة لكل التطبيق بمعنى إذا دخل مستخدم (أ) و قام بكتابة كلمة و دخل مستخدم (ب) فسوف يجد الكلمة التى كتبها المستخدم (أ) و طبعا هذا غير صحيح بالنسبة للحفاظ على خصوصية المستخدمين لذلك سوف نستخدم **User** لأنه مخصصة للمستخدم بحد ذاته و لكن كيف سيفرق التطبيق بين المستخدمين هذا ما سوف يتم شرحه قريبا فى الفصل الخاص بالأمان و المستخدمين لنعد مرة أخرى الى موضوعنا، بعد هذا المثال الطويل سوف نبدأ رحلة جديدة مع أدوات جديدة و طبعا كلها لها علاقة بـ **WebParts**.

و الآن لنضيف بعض الأدوات و التى سوف تعطي للمستخدم أماكنات جديدة و بالطبع سوف يكون هذا بأقل جهد ممكن و الأدوات التى سوف نضيفها هما كالتالى

- **CatalogZone**

هذه الاداه تحتوى بعض الأدوات (**PageCatalogPart / DeclarativeCatalogPart**)

هذه الاداة تقوم بعمل قائمة بالأدوات المتاحة و إتاحة أماكنات الأدوات الى هذه الأدوات على سبيل يمكن للمستخدم اظهار و أخفاء ما يري من أجزاء ويب بالإضافة لاي أستيراد خصائص الأدوات التعريف يبدو غير واضح ولكن سوف يتضح أكثر مع المثال.

- **DeclarativeCatalogPart**

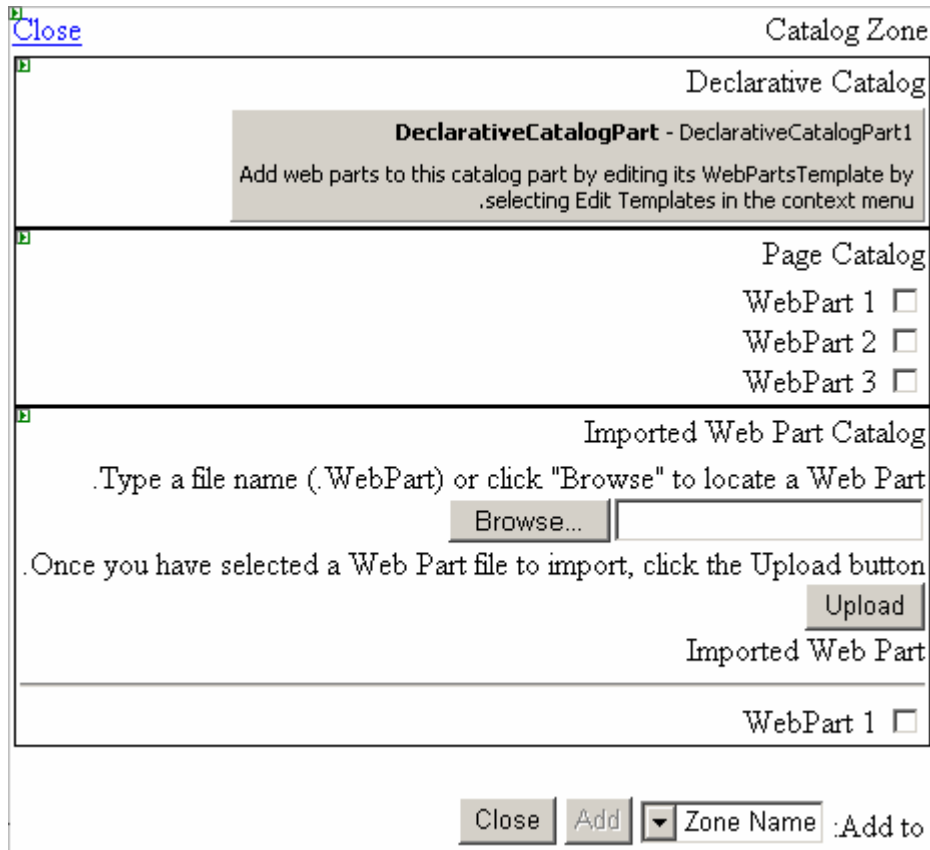
تتيح للمطور عارض قائمة بالأدوات التى يمكن التعامل معها (إضافة و حذف فى أجزاء الويب **WebPart zone**)

- **PageCatalogPart**

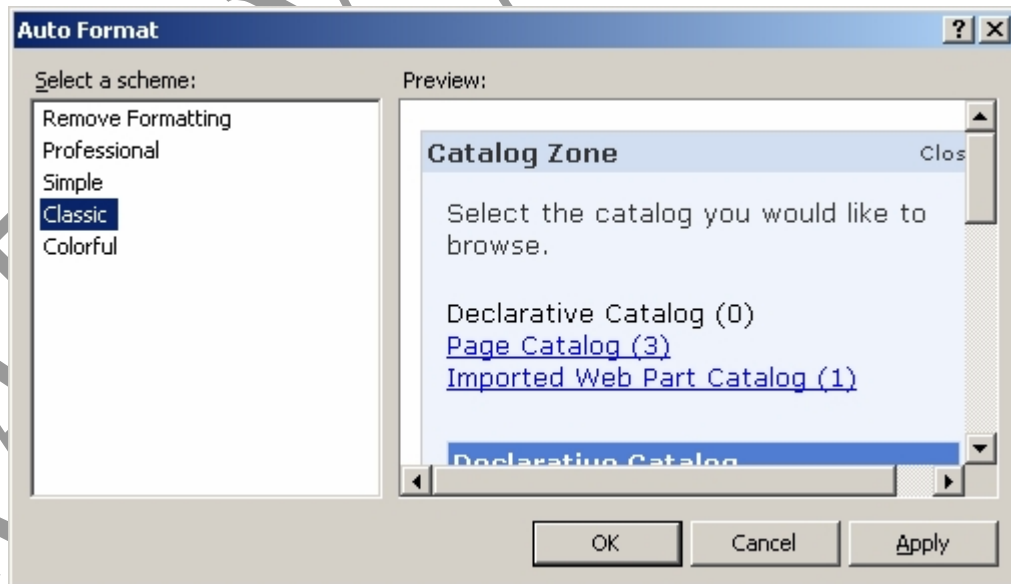
عرض صفحة بالادوات التى تعامل معها المستخدم و قام بأغلاقها للتيح له من جديد إضافة هذه الأدوات.

- **ImportCatalogPart**

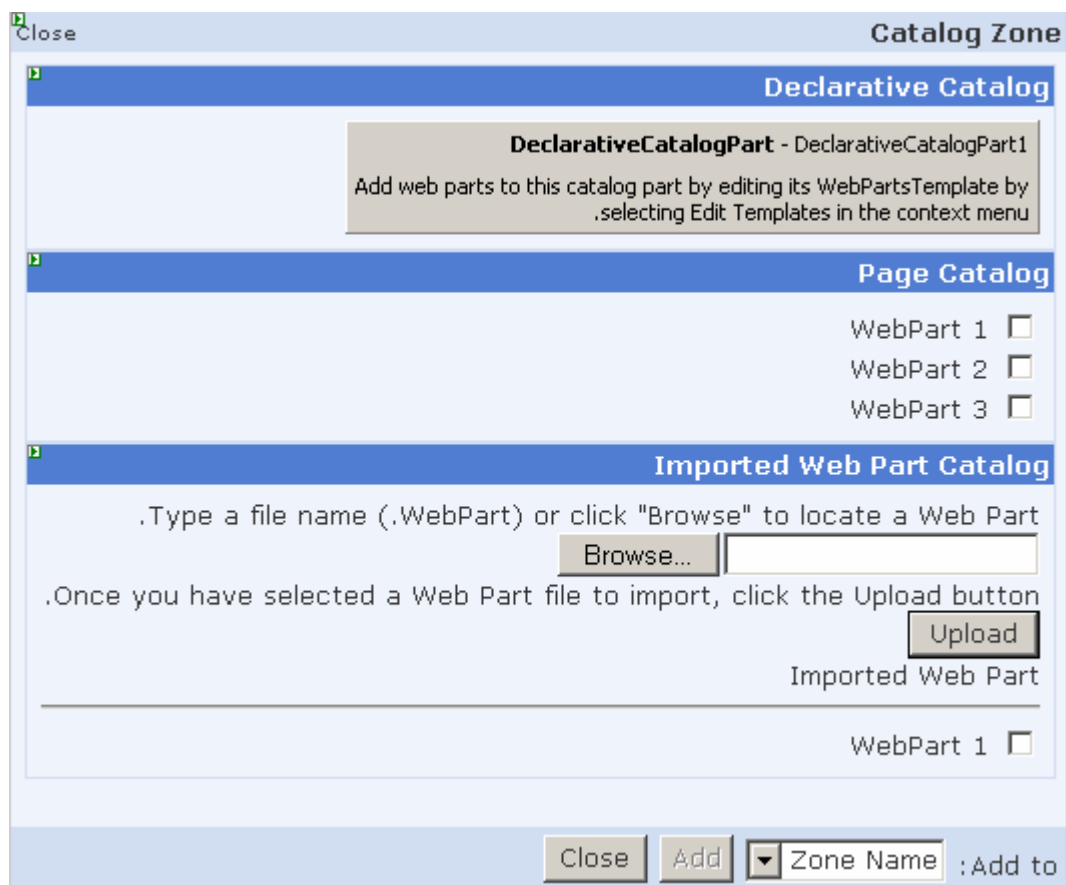
هذه الاداة تقوم بأستيراد خصائص الاداة عن طريق ملف الذى قمنا بحفظه من قبل.



الشكل 43 كما نرى أن الاداة CatalogZone تحتوي على بقية الأدوات – إذا حاولت وضع هذه الأدوات خارج CatalogZone لن تعمل هذه الأدوات.



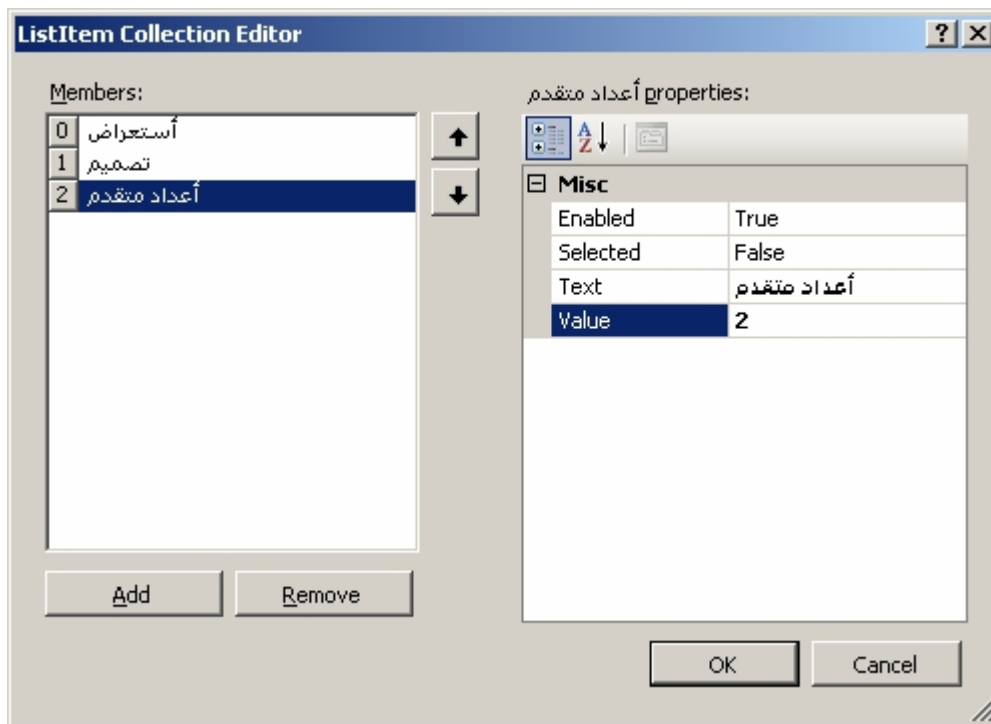
الشكل 44 فقط للتذكير أن تغيير الشكل قد يكون مهم و مريح نفسيا J



الشكل 45 هذه هو الشكل بعد تغيير Style

و الآن لنرى كيف تعمل هذه الأدوات سوياً

أول خطوة نقوم بأضافة 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
```

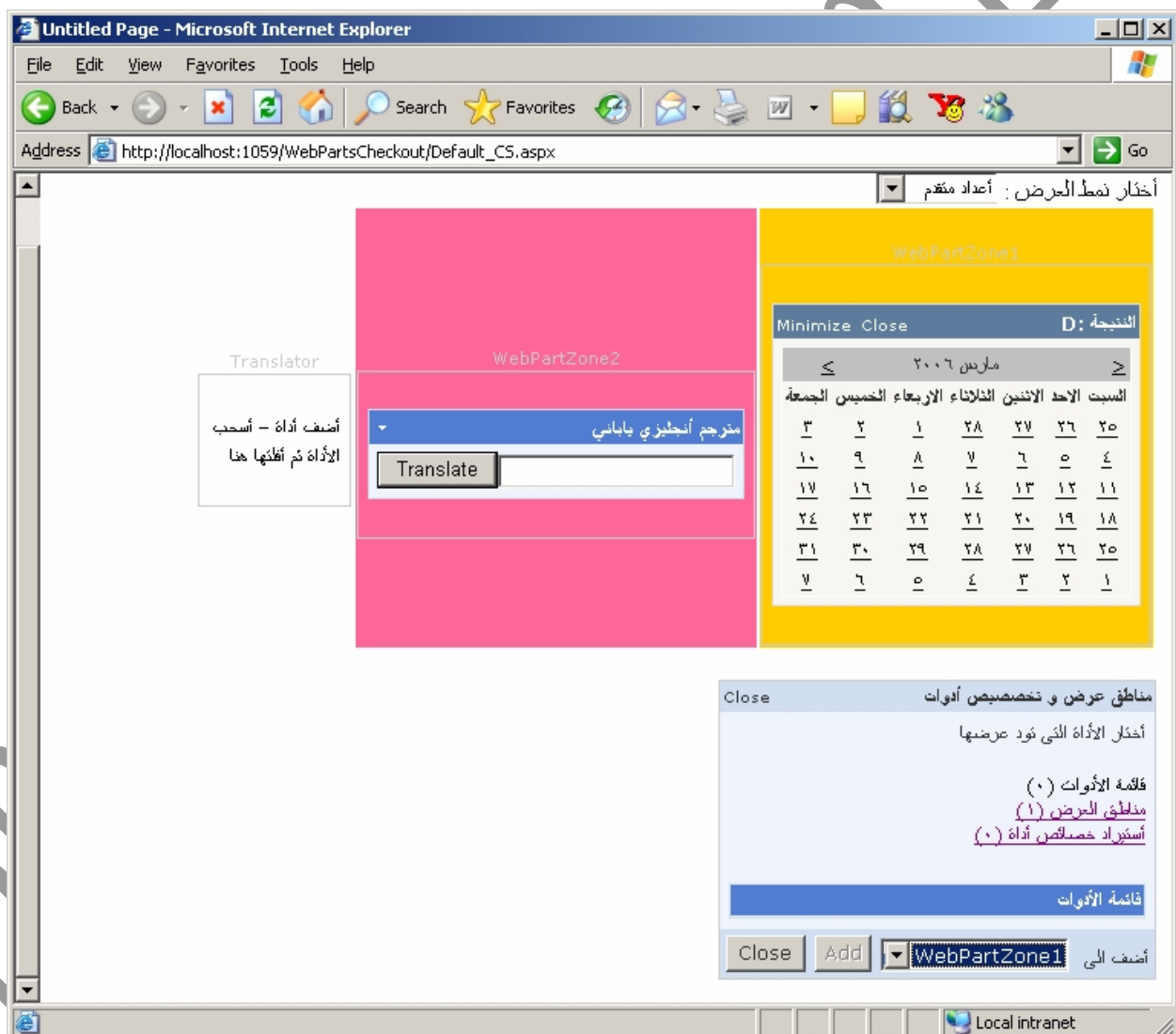
Case Else

```
WebPartManager1.DisplayMode = WebPartManager.BrowseDisplayMode
```

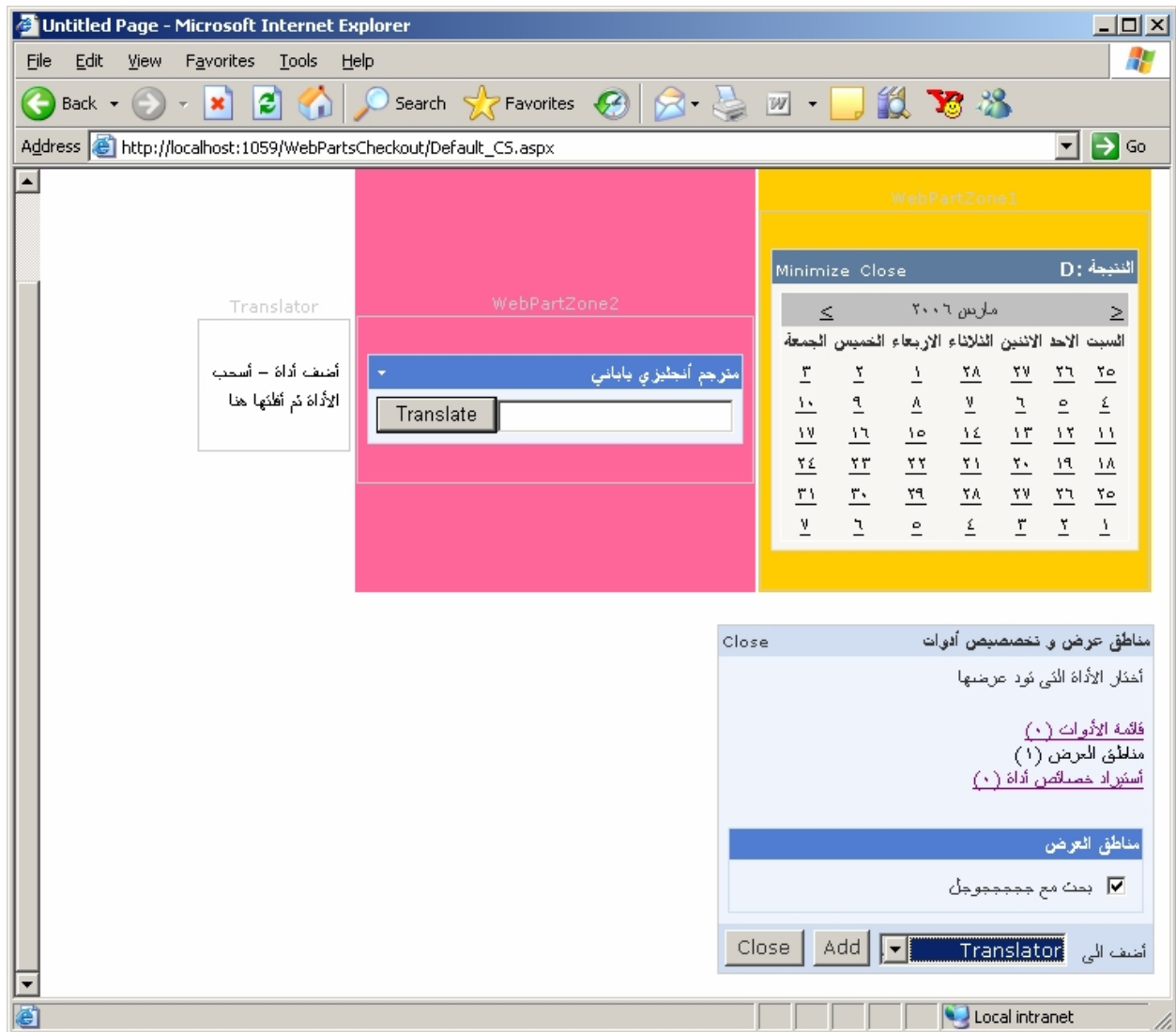
End Select

End Sub

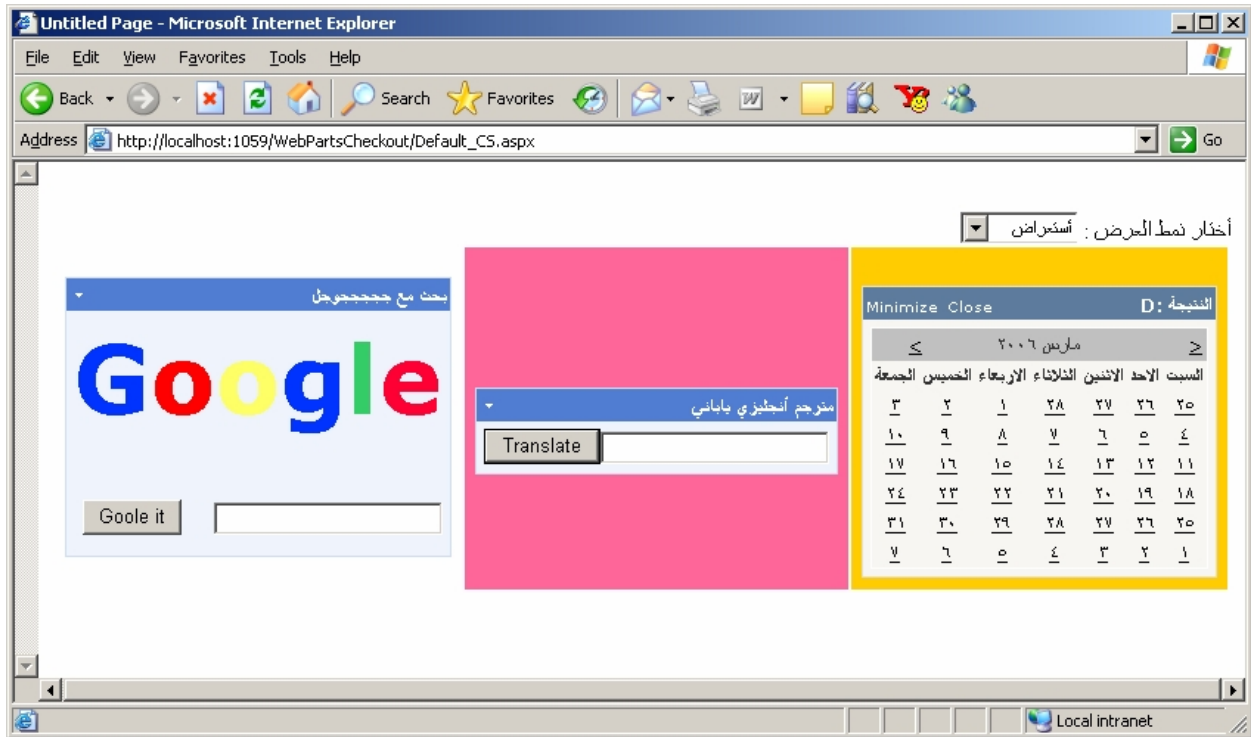
الكود بصيغة VB.net



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



الشكل 48 تحديد الأداة و مكان العرض



الشكل 49 بعد ذلك بمجرد العودة الى 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 و كأن المستخدم يعمل على الـ VS بمعنى أن المستخدم سوف يكون قادر على تغيير كل خصائص العناصر الموجودة في الصفحة كالعادة يبدو الكلام غير واضح J و لكن لاداعي للقلق لانه هناك مثال لتوضيح هذه النقطة و الآن لنكمل على المثال. سوف نضيف بعض الأدوات للمثال

- EditorZone

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

- **PropertyGridEditorPart**

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

- **LayoutEditorPart**

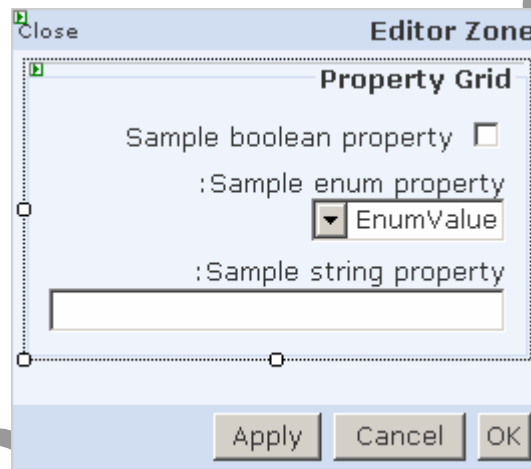
تعطي المستخدم القدرة على تحديد مكان عرض الأداة بالإضافة الى بعض خصائص العرض.

- **AppearanceEditorPart**

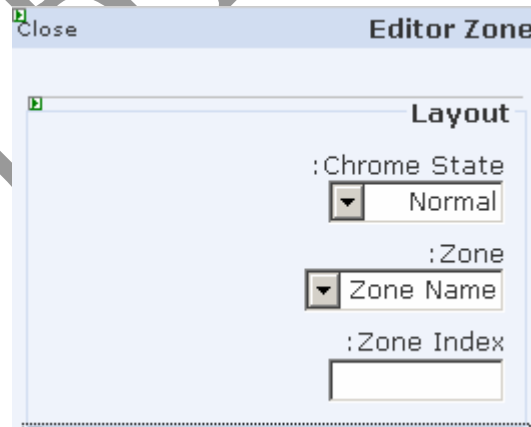
هذه الأداة القدرة على تعديل الأداة بشكل رائع مع القدرة على تحديد التفاصيل بدقة

- **BehaviorEditorPart**

تعطي سيطرة على خصائص أكثر دقة و سوف يتضح في المثال



الشكل 51



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 ☐

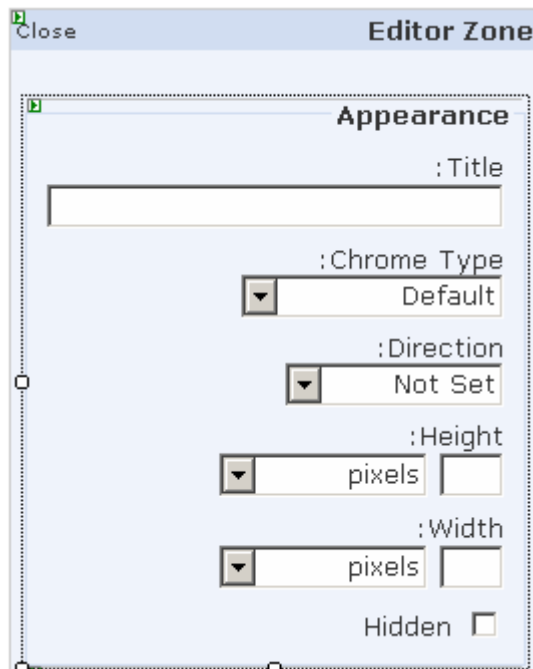
Allow Edit ☐

Allow Hide ☐

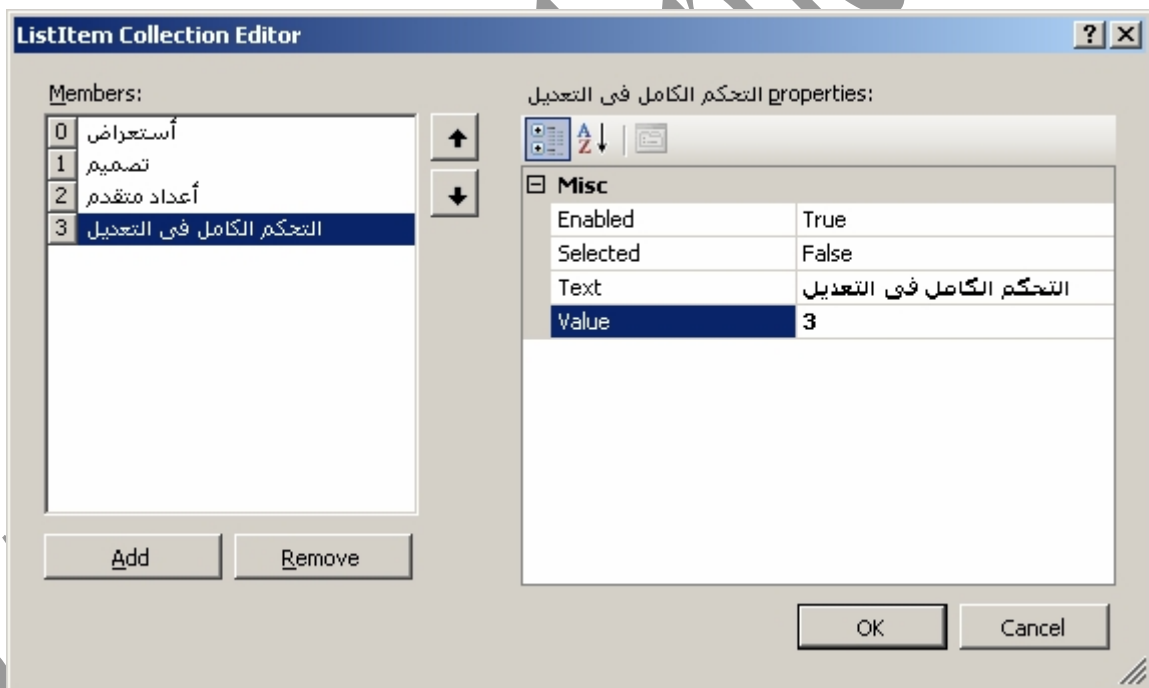
Allow Minimize ☐

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;
}
}

```

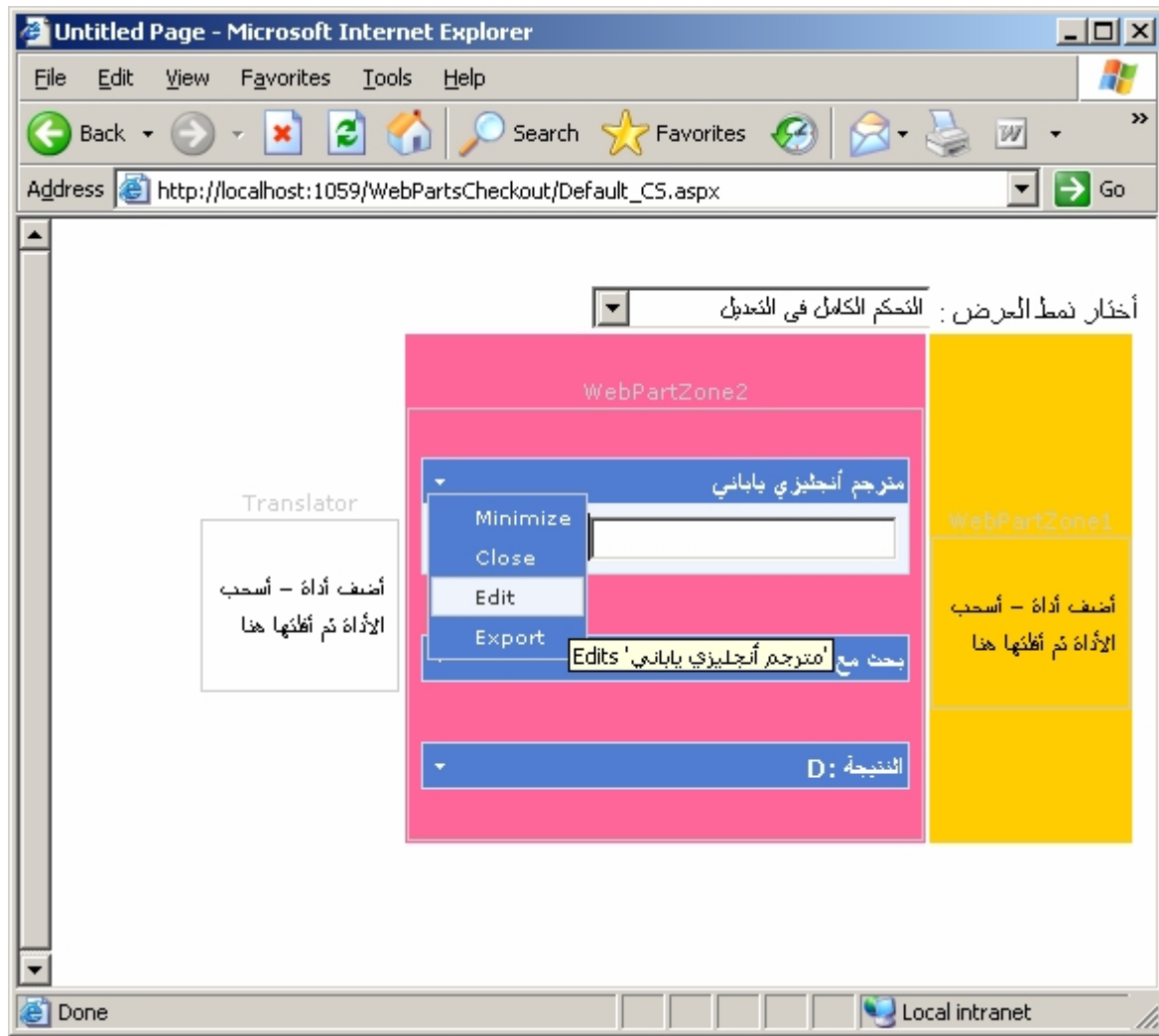
الكود بصيغة الـ C#

```

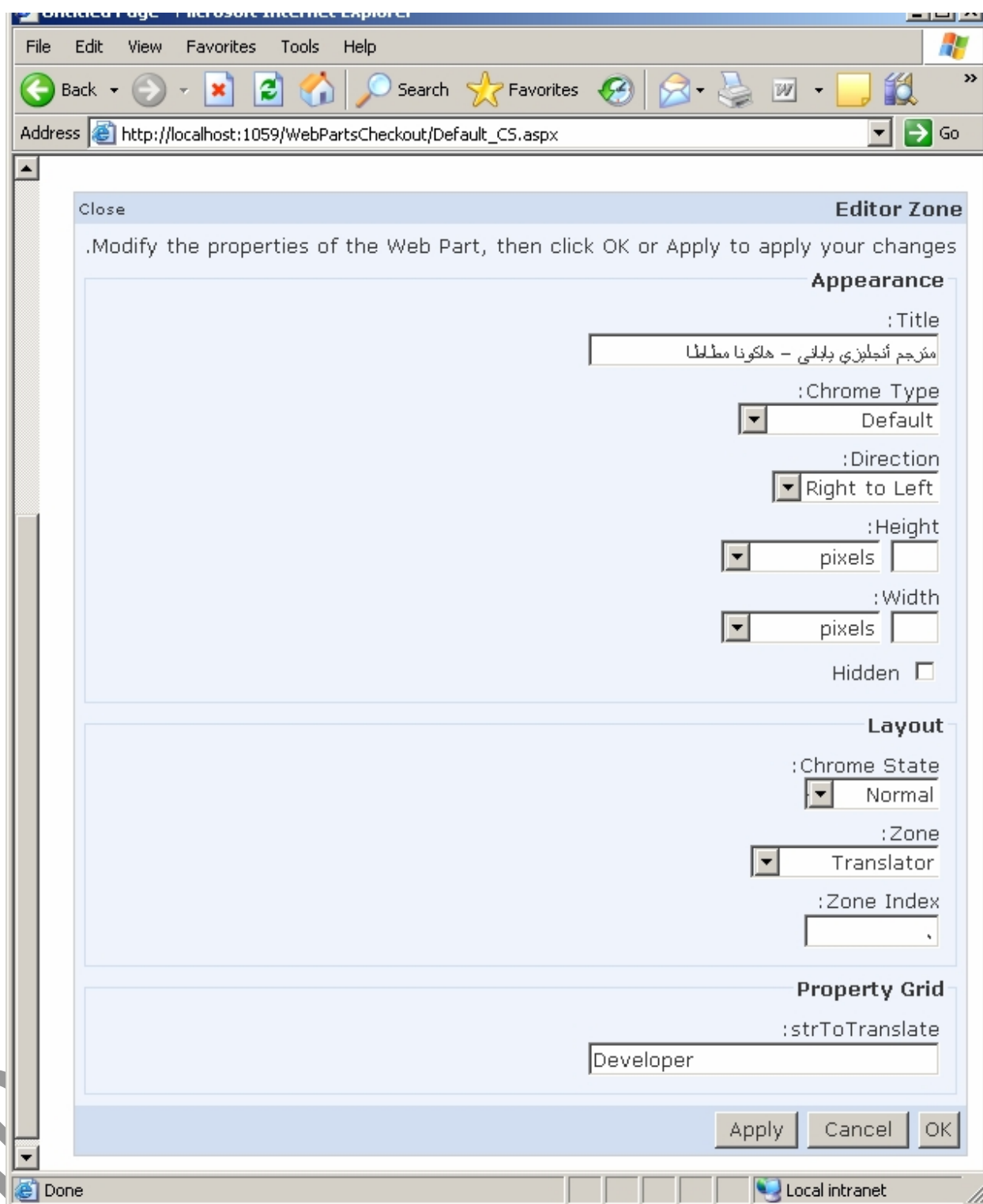
Protected Sub DropDownList1_SelectedIndexChanged(ByVal sender As Object,
ByVal e As System.EventArgs) Handles ddl_ShowType.SelectedIndexChanged
    Select Case Val(ddl_ShowType.SelectedValue)
        Case 0
            WebPartManager1.DisplayMode = WebPartManager.BrowseDisplayMode
        Case 1
   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

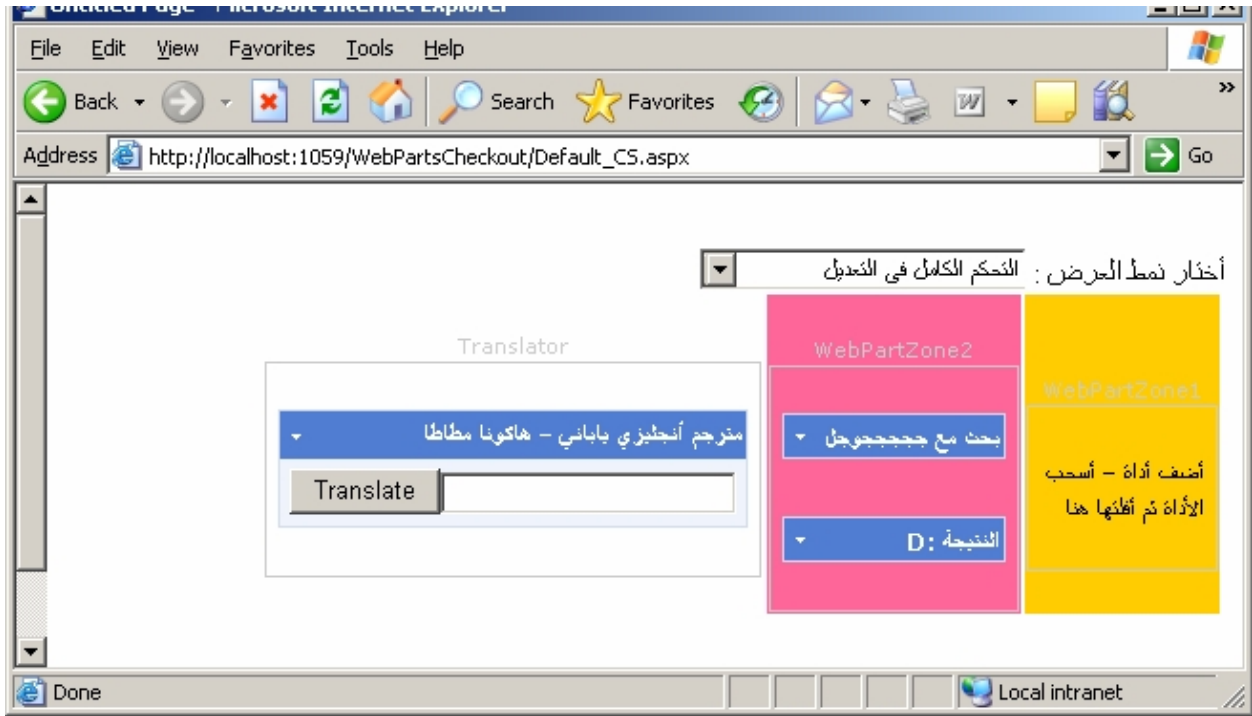


الشكل ٥٦ : للتعامل مع إمكانية التعديل لابد من اختيار **Edit** من قائمة كل أداة سوف تظهر القائمة **Edit** لكل أداة بعد تغيير نمط العرض الى "التحكم الكامل في التعديل"



الشكل 57 كما تلاحظ أنه يمكن تغيير أي خاصية لأي أداة كل ما عليك بعد تعديل أضغط OK

هناك معلومة صغيرة و هي أنه يمكنك ترجمة كل الكلمات الموجودة في الأدوات و كل هذا من نافذة الخصائص **Properties window**



الشكل 58 هذا هو الشكل النهائي.

و في النهاية هذا الفصل أود أن نذكر سويا ملخصا لما تعلمناه في هذا الفصل:

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