الفصل السادس عشر

تجميع تطبيقات الويب المكونة من ثلاث طبقات

ASSEMBLING A THREE-TIER WEB FORM APPLICATION

أهداف الفصل:

- التعرف على مفهوم الحالة (State) في تطبيقات الويب.
- إنشاء عنصر واجهة الاستخدام (ASP.NET User Control).
- استخدام تقنية ربط البيانات (Data Binding).
- تطوير تطبيق ويب لشركة برادشو مارينا.
- التعرف على كل من لغة XML وخدمات الويب Web Services.

عزيزي القارئ سوف تتعلم خلال الفصل الأخير من هذا الكتاب على تقنية تطوير تطبيق الويب الذي يتكون من ثلاثة أنواع من الأصناف لشركة برادشو مارينا، حيث عرضنا خلال الفصل الثاني عشر المفاهيم الأولية لتطوير تطبيقات الويب باستخدام تقنية ASP.NET، كما قدمنا خلال كل من الفصل الثالث عشر والفصل الرابع عشر تفاصيل تخزين الكائنات والتعامل مع الملفات وإنشاء أصناف مجال المشكلة لتطوير تطبيقات تتكون من ثلاث طبقات من الأصناف حيث ركزنا خلال الفصل الخامس عشر على تطوير تطبيقات الويندوز، وسوف نركز خلال هذا الفصل على تطبيقات الويب.بالرغم من أن الفصل الثاني عشر قدم مفاهيم هامة عن تقنية ASP.NET، إلا أننا نحتاج إلى اكتشاف العديد من المفاهيم والمميزات قبل تطوير تطبيق الويب الخاص بشركة برادشو مارينا.

سوف نتعلم خلال هذا الفصل مفهوم الحالة (State) المرتبط بشبكة الإنترنت، كما سنستخدم عناصر ASP.NET User Controls لتحسين تطبيق شركة برادشو مارينا، مع العلم أن تقنية ASP.NET تقدم إمكانية ربط البيانات (Data Binding) التي تربط عنصر واجهة استخدام بمصدر للبيانات، وسوف نعتمد على هذه التقنية في تطوير تطبيق شركة برادشو مارينا.

التعرف على مفهوم الحالة (State) لتطبيقات الويب

Understanding the Concept of State for Web Applications

تعتمد خدمة الويب على بروتوكول HTTP وهو اختصار للجملة Hypertext Transmission Protocol والمعروف بعدم تذكره لحالة المستخدم (Stateless Protocol)، ولقد رأينا خلال الفصل الثاني عشر عندما يطلب المستخدم صفحة ويب من خادم الويب ترسل رسالة إلى خادم الويب بواسطة الكائن HTTP Request طالباً منه صفحة محددة، فإن وجد خادم الويب الصفحة المطلوبة يقوم بمعالجة البيانات المطلوبة وإنشاء لغة HTML المناظرة ثم إرسالها إلى عميل الويب بواسطة الكائن HTTP Response. ولأن خادم الويب لا يعلم هل طلب الصفحة هذه يمثل جزءاً من عدة طلبات من المستخدم نفسه أو طلب واحد منفصل فإن برتوكول HTTP يطلق عليه Stateless Protocol.

لقد صمم برتوكول HTTP لكي يتصل خادم الويب بعميل الويب دون الاحتفاظ بحالة عميل الويب بغرض زيادة استيعاب عدد أكبر من عملاء الويب الذين يتعاملون في نفس اللحظة مع خادم الويب، أما الاحتفاظ بحالة المستخدم ووجود اتصال مستمر بينهم أثناء تصفح المستخدم لصفحات الموقع فيتطلب ذلك موارد كثيرة مما يؤدي إلى إضافة عبء على خادم الويب؛ ولذلك فإن برتوكول HTTP يصل خادم الويب بعميل الويب عندما يريد المتصفح استدعاء صفحة ويب فقط من الخادم، ولكن على أي حال فإن تطوير التطبيقات في بيئة مثل تلك تكون أكثر تعقيداً من بيئات أخرى مثل بيئة الويندوز التي توفر اتصالاً دائماً بين جهاز العميل وجهاز الخادم، وسوف نوضح خلال الفقرات التالية كيف يتم الاحتفاظ بحالة العميل بواسطة تقنية ASP التقليدية (استخدام الكائن Application والكائن Session وملفات Cookies) ثم نرى كيف يتم الاحتفاظ بحالة المستخدم بواسطة ASP.NET.

الاحتفاظ بحالة العميل

Maintaining State

ولأن برتوكول HTTP لا يحتفظ بحالة عميل الويب فإنه توجد أكثر من طريقة للاحتفاظ بحالة العميل:

- **حالة التطبيق**: يتم ذلك من خلال تعريف متغيرات داخل الكائن Application لتكون مشتركة على مستوى جميع متصفحي الموقع.
- **حالة الجلسة**: يتم ذلك من خلال تعريف متغيرات داخل الكائن Session لتكون خاصة بكل مستخدم على حده طوال فترة تصفحه لصفحات الموقع.
- **ملفات Cookies**: الاحتفاظ بمعلومات قليلة (أقل من ٤ كيلو بايت) داخل ملفات على جهاز المستخدم التي تبقى صالحة لفترة زمنية محددة.

- قواعد البيانات: الاحتفاظ بكم هائل من المعلومات داخل قاعدة بيانات التي تبقى مع الخادم حتى بعد انتهاء جلسات المستخدم.

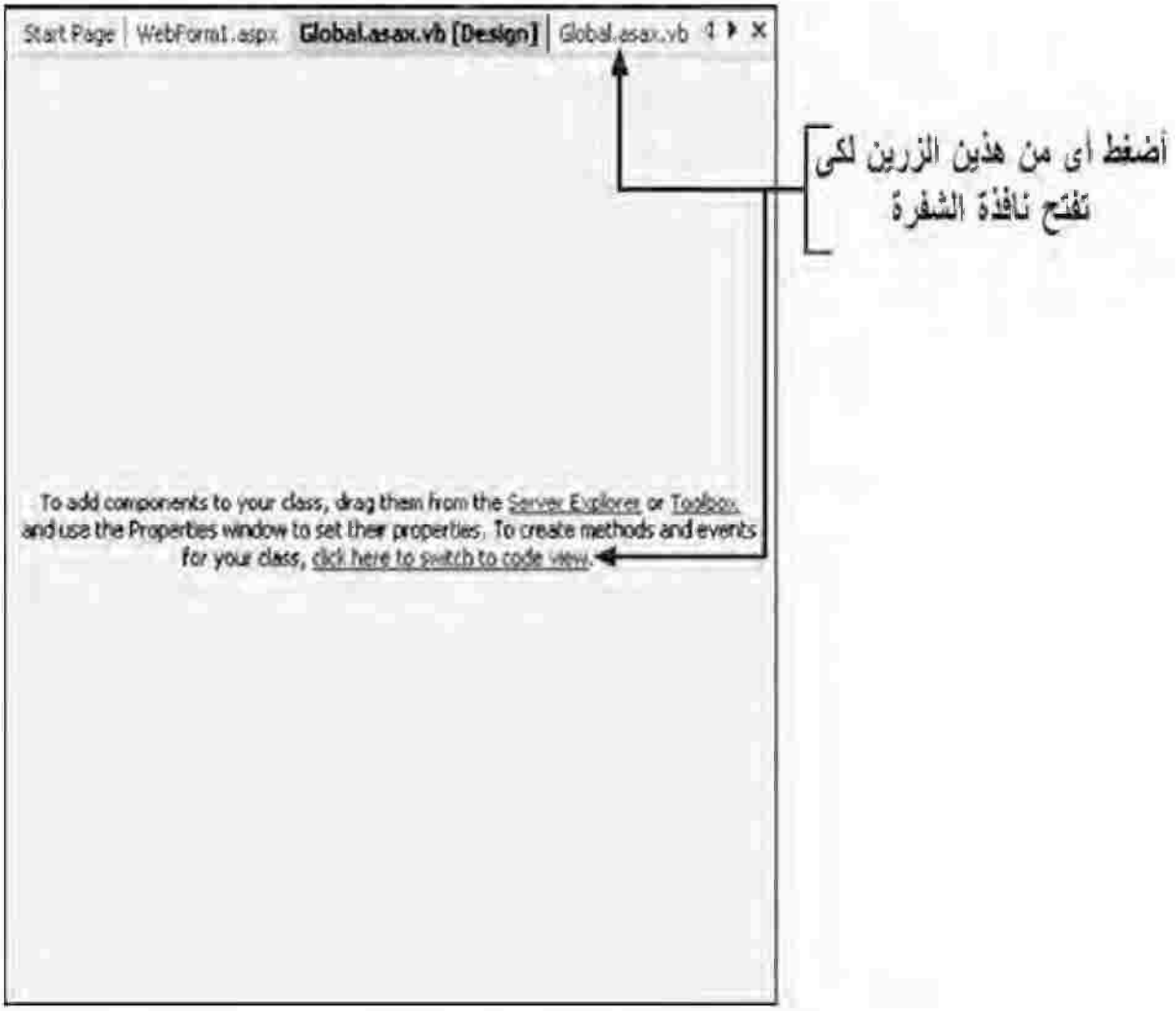
يوجد كائن من الصنف Application لكل موقع إنترنت حيث يحفظ بمعلومات مشتركة على مستوى جميع المستخدمين، كما يتم إنشاء كائن من النوع Session لكل مستخدم حيث يتم تخزين معلومات كل مستخدم على حده وتبقى المعلومات أثناء تواجد المستخدم ويتم حذفها عند مغادرة المستخدم للموقع. وتستطيع تقنية ASP.NET أيضاً استخدام مفهوم الجلسة (Session) للاحتفاظ بحالة المستخدم دون الاعتماد على ملفات Cookies خاصة به.

التعرف على حالة التطبيق وكائن Application

Understanding Application State and the Application Object

يقدم كائن Application وسيلة هامة لتخزين معلومات عامة على مستوى جميع المستخدمين لنفس تطبيق الويب حيث تقوم تقنية ASP.NET بتخزين متغيرات الكائن Application بشكل مركزي حيث يمكن الوصول لهذه المتغيرات عن طريق الصفة Application للصنف Page، كما تقدم تقنية ASP.NET ملف Global.asax داخل كل تطبيق ويب والذي يمثل مرحلة البداية لتعريف قيم أولية لمتغيرات الكائن Application وإستخدامها. يوضح الشكل رقم (١٦،١) هذا الملف داخل نافذة مستكشف الحلول، أما الشكل رقم (١٦،٢) فيوضح نافذة نموذج الويب الناتج لملف Global.asax.

الشكل رقم (١٦،١). ملف Global.asax.



الشكل رقم (١٦,٢). رؤية تصميم Global.asax.vb.

كما يوضح الشكل رقم (١٦.٣) شفرة ملف Global.asax الذي يحتوي على إجراءات معالجة أحداث بداية جلسة ونهايتها وأحداث بداية التطبيق ونهايته.

ولكي نوضح كيفية استخدام إجراء معالجة حدث بداية التطبيق Application_Start()، دعنا نعرف متغيراً لتخزين رقم هاتف شركة برادشو مارينا كمتغير على مستوى التطبيق، ودعنا أيضاً نعرف كائناً من الصنف Boat كمتغير على مستوى التطبيق داخل هذا الإجراء، ولكي نكمل هذا المثال يجب علينا أن نضيف الصنف Boat من الفصل التاسع إلى المشروع الحالي، كما يجب أن نضيف أربع صفات للقراءة فقط داخل هذا الصنف: الصفة strRegNo، والصفة length، والصفة manuf، والصفة btYear، وبالرغم من أننا يمكن أن نستخدم إجراءات المرور بدلاً من تلك الصفات إلا أنه من المفضل تعريف تلك الخصائص لأن ربط البيانات ببعض عناصر خادم الويب

تتطلب تعريف خصائص عامة. توضح الأوامر التالية تعريف إجراء معالجة حدث بداية تعريف التطبيق Application_Start حيث يبدأ الإجراء بتعريف المتغير BradshawMarinaPhone لتخزين رقم هاتف شركة برادشو مارينا، ويعرف الأمر الثاني كائناً من الصنف Boat وإسناده إلى المتغير النصي BoatState، ويجب أن نلاحظ أن متغيرات التطبيق يتم تعريفها مرة واحدة عند بداية تطبيق الويب.

```
Sub Application_Start(ByVal sender As Object, _
   ByVal e As EventArgs)
   ' Fires when the application is started

   Application("BradshawMarinaPhone") = "840-123-5432"
   Application("boatState") = New _
          Boat("MO1234567", 24, "Ranger", 2002)
End Sub
```

الشكل رقم (١٦,٣). رؤية شاشة Global.asax.vb.

ولكي نوضح كيف نستخدم متغيرات التطبيق داخل تطبيق الويب، قم بتصميم صفحة ويب تحتوي على عنوان من النوع Web Server لكي تظهر خصائص المركب، فعلى سبيل المثال يمكنك استخدام الأسماء lblPhone و lblStRegNo و lblLength و lblManuf و lblYear في تسمية هذه العناوين، كما يمكنك إضافة زرين (الزر "Display" والزر "New Phone No").

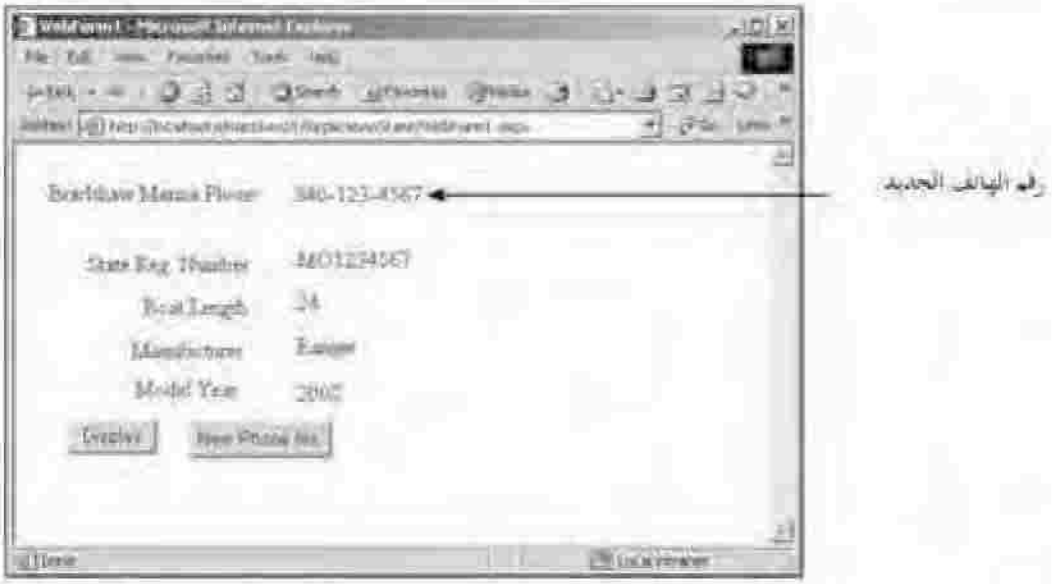
يستطيع مستخدم هذه الصفحة الضغط على زر "Display" لعرض بيانات المركب الأولية، كما يمكنه الضغط على الزر "New Phone No" لتغيير رقم هاتفه وعرضه على الصفحة. يوضح الإجراء التالي أوامر الزر "Display" حيث يستخلص الأمر الأول قيمة متغير التطبيق BradshawMarinaPhone وإسناده للخاصية Text للعنوان lblphone لعرض رقم الهاتف، ثم ينشئ الأمر الثاني كائناً من الصنف Boat (aBoat) بتحويل متغير التطبيق النصي "BoatState" إلى النوع Boat، ثم تسترجع الأوامر الأربعة التالية قيم خصائص القراءة فقط المعرفة داخل الصنف وإسنادها إلى الخاصية Text للعناوين الأربعة المذكورة سابقاً وعرضها على صفحة الويب كما هو واضح في الشكل رقم (١٦,٤).

```
Private Sub btnShowBoat_Click(ByVal sender As System.Object, _
     ByVal e As System.EventArgs) Handles btnShowBoat.Click

     ' Put Bradshaw Marina phone number on a label
     lblPhone.Text = Application("BradshawMarinaPhone")

     ' Create a boat instance from the applicatin string
     Dim aBoat As Boat = CType(Application("boatState"), Boat)

     ' Put boat information on labels
     lblStReg.Text = aBoat.stateRegNo
     lblLength.Text = aBoat.lngth
     lblManuf.Text = aBoat.manuf
     lblYear.Text = aBoat.btYear

End Sub

Private Sub btnNewPhoneNo_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnNewPhoneNo.Click
    Application.Lock()
    Application("BradshawMarinaPhone") = "840-123-4567"
    Application.UnLock()
    lblPhone.Text = Application("BradshawMarinaPhone")
End Sub
```

ولأن متغيرات التطبيق عامة على مستوى جميع المستخدمين فربما تواجه تعارضاً إذا حاول أكثر من مستخدم تغيير متغير تطبيق في الوقت نفسه؛ ولذلك يجب استدعاء الإجراء Lock قبل تعديل قيمة متغير تطبيق

واستدعاء الإجراء Unlock بعد تعديل قيمة متغير تطبيق، ويوضح إجراء معالجة الزر "New Phone No" كيف يتم استخدام كل من الإجراء Lock والإجراء Unlock لتعديل رقم هاتف شركة برادشو ماريتا، ثم عرض رقم الهاتف الجديد كما هو واضح في الشكل رقم (١٦،٥) بعد تغيير الزر "New Phone No" بعد الضغط على زر "Display".

الشكل رقم (١٦،٤). قيم متغير التطبيق.



الشكل رقم (١٦،٥). رقم هاتف جديد متغير التطبيق.

بالرغم من إمكانية تخزين كائنات الصنف Boat كمتغيرات تطبيق إلا أن ذلك يقلل من كفاءة تطبيق الويب ؛ ولذلك من المفضل أن يتم تخزين كائنات الصنف Boat داخل قاعدة البيانات ، وتذكر أن تطبيق الويب يحتوي على كائن Application واحد فقط ، ولكن على أي حال إن المثال السابق الذي أسند كائناً من الصنف Boat إلى متغير تطبيق واستخدام متغير تطبيق آخر لتخزين رقم هاتف شركة برادشو مارينا يوضح كيفية استخدام متغيرات التطبيق.

التعرف على حالة الجلسة وكائن الجلسة

Understanding Session State and the Session Object

يقدم كائن الجلسة (Session Object) أسلوباً جيداً للاحتفاظ بمعلومات كل مستخدم على مستوى جميع صفحات تطبيق الويب ، حيث تبدأ الجلسة عند زيارة المستخدم للموقع وتنتهي عند مغادرة المستخدم الموقع أو عند انتهاء وقت الجلسة (الوقت الذي يمر دون تفاعل بين المستخدم والموقع) ، مع العلم أن وقت الجلسة الافتراضية هو ٢٠ دقيقة. تعتمد معظم مواقع التجارة الإلكترونية ومواقع سلة التسوق (Shopping Cart) على بيانات الجلسة حيث تحتفظ بطلبات شراء كل مستخدم داخل الجلسة الخاصة به ، كما يمكنك استخدام الجلسة لتغيير مظهر صفحات الموقع بناء على رغباته (Personalization) ، ويوضح الشكل رقم (١٦,٣) أوامر إجراء كل من معالجة حدث عند بداية ونهاية الجلسة.

فعلى سبيل المثال إذا افترضنا أننا نريد تطوير سلة التسوق بسيطة حيث يتم زيادة عدد المنتج بمقدار واحد عند الضغط على زر ما ، ومسح محتويات سلة التسوق عند الضغط على زر آخر. فإذا افترضنا أن شركة برادشو مارينا تقوم ببيع هدايا تذكارية ، فسنحتاج إلى إضافة سلة التسوق لتحتفظ ببيانات العناصر المباعة ؛ ولذلك سيتم تعريف متغير جلسة يحتفظ بعدد عناصر سلة التسوق.

توضح الأوامر التالية إجراء معالجة بداية جلسة جديدة Session_Start() حيث يقوم بتعريف متغير الجلسة "NbrInCart" الذي سيحتفظ بعدد العناصر المباعة وإسناد قيمة ابتدائية (صفر) له.

```
Sub Session_Start(ByVal sender As Object, _
     ByVal e As EventArgs)
     ' Fires when the session is started

     Session("NbrInCart") = 0
End Sub
```

تحتوي صفحة ويب هذا المثال على زرين وهما : الزر "Add to shopping cart" لإضافة القيمة واحد لعدد العناصر المباعة ، والزر "Clear shopping cart" لمسح محتويات سلة التسوق ، كما تحتوي على العنوان lblCount الذي يظهر عدد العناصر المباعة الحالية. توضح الأوامر التالية ثلاثة إجراءات وهي : الإجراء Page_Laod المسؤول عن

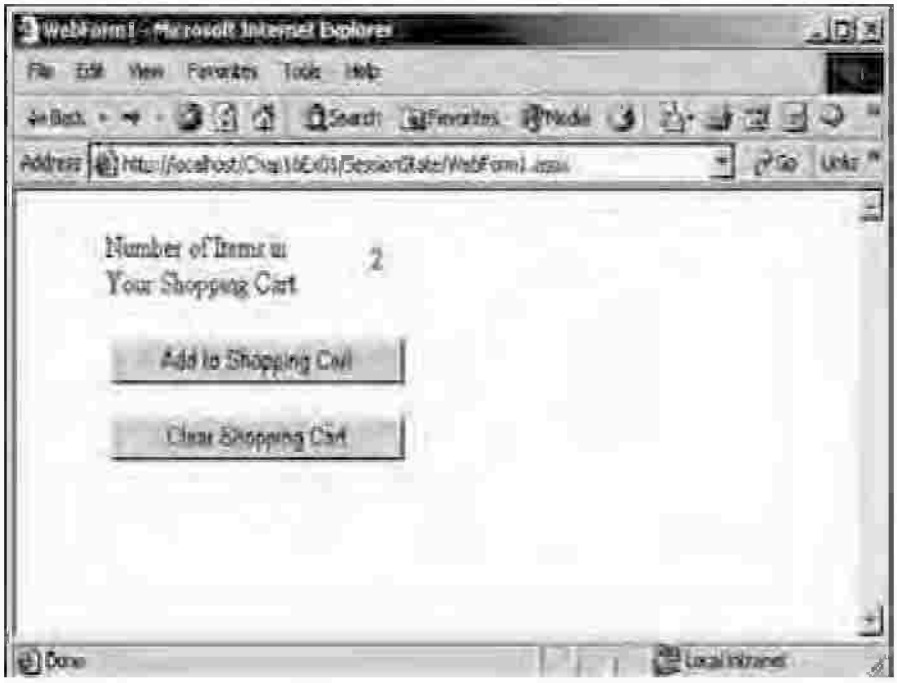
إسناد قيمة متغير الجلسة NbrInCart إلى الخاصية Text للعنوان lblCount عند بداية تحميل الصفحة، والإجراء btnAdd_Click الذي يضيف القيمة واحد إلى متغير الجلسة NbrInCart ثم إظهاره على العنوان، والإجراء btnClear_Click لإسناد قيمة صفر للمتغير NbrInCart ثم إظهاره على العنوان lbCcount.

```
Private Sub Page_Load(ByVal sender As System.Object, _
     ByVal e As System.EventArgs) Handles MyBase.Load
     'Put user code to initialize the page here

     lblCount.Text = Session("NbrInCart")
End Sub

Private Sub btnAdd_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnAdd.Click

    Session("NbrInCart") += 1
    lblCount.Text = Session("NbrInCart")
End Sub

Private Sub btnClear_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnClear.Click

    Session("NbrInCart") = 0
    lblCount.Text = Session("NbrInCart")
End Sub
```

يوضح الشكل رقم (١٦,٦) صفحة سلة التسوق لشركة برادشو مارينا، ولاحظ أن المستخدم سيفقد أي بيانات مخزنة داخل بيانات الجلسة بعد مرور زمن الجلسة؛ ولذلك يمكن الاستعانة بملفات Cookies للاحتفاظ بالبيانات مرة أطول والتي سوف نناقشها في الفقرة التالية.

استخدام ملفات Cookies
Using Cookies

إن ملف Cookie عبارة عن ملف نصي يتم تخزينه على جهازك بواسطة تطبيق الويب الذي تقوم بزيارته، وعادة يحتوي ملف Cookie على اختيارات المستخدم أو على بيانات دخول المستخدم للموقع، وبشكل عام إن ملفات Cookies تقدم وسيلة جيدة للاحتفاظ بحالة المستخدم حتى بعد انتهاء زيارته للموقع. يسمح برنامج متصفح الإنترنت باستقبال ملفات Cookies وتخزينها على جهازك، ولكن يمكنك تغيير هذه الاختيارات داخل متصفح الويب الخاص بك لمنع استقبال هذه الملفات وتخزينها. ويستطيع تطبيق الويب تحديد تاريخ انتهاء صلاحية ملف Cookies أو سيتم انتهاء صلاحية الملف بعد انتهاء الجلسة مباشرة.

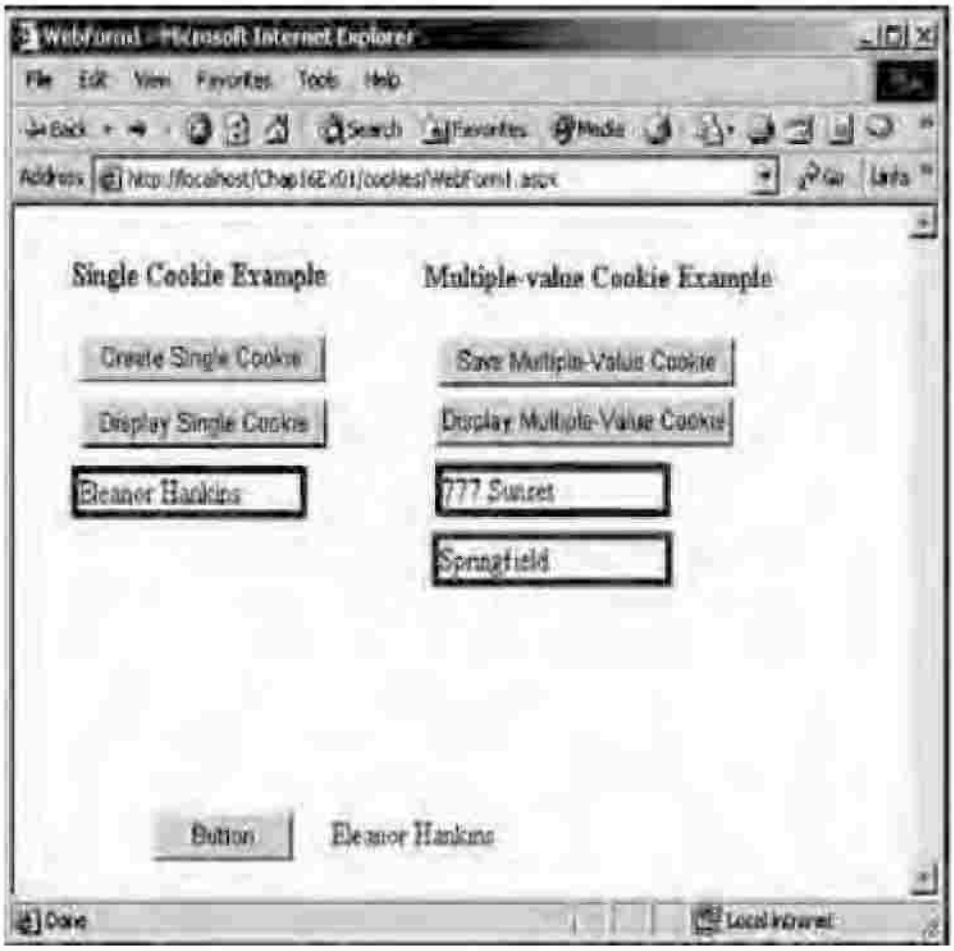


الشكل رقم (١٦,٦). تطبيق ويب سلة التسوق.

يمكن أن يحتوي ملف Cookie على قيمة مفردة أو على عدة قيم. وكما هو واضح في الشكل رقم (١٦,٧) يحتوي الإطار الأيسر لنموذج الويب على زرين، فإذا ضغط المستخدم على الزر "Create Single Cookie" فسوف يتم إنشاء ملف Cookie يحتوي على قيمة مفردة لم يسند له تاريخ صلاحية، أما إذا ضغط المستخدم على الزر "Display Single Cookie" فسيتم استرجاع بيانات ملف Cookie وعرضها داخل العنوان الذي يظهر أسفل الزرين.

أما إطار نموذج الويب الأيمن فيحتوي على زرين وعنوانين، ومثل الإطار الأيسر إذا ضغط المستخدم على الزر "Save Multiple-Value Cookie" فيتم إنشاء ملف Cookie ذو قيم متعددة ثم يتم إسناد تاريخ انتهاء صلاحية له، وأما إذا ضغط المستخدم على الزر "Display Multiple-Value Cookie" فيتم استرجاع قيم ملف Cookie الذي تم إنشاؤه سابقاً وعرضها داخل العناوين المتواجدة أسفل الزرين، ويوضح الشكل رقم (١٦,٨) إجراءات معالجة الأزرار الأربعة.

يتم كتابة بيانات ملفات Cookie باستخدام الكائن Response واسترجاع قيم ملف Cookie باستخدام الكائن Request، وكما هو واضح من أوامر الإجراءات السابقة فإن اسم ملف Cookie ذو القيمة الواحدة هو CustomerName وبما أن تقنية ASP.NET مثل لغة VB .NET تدعم استدعاء نالفة الأوامر الذكية (IntelliSense)، فعند كتابة نقطة بعد اسم الكائن Response تظهر نالفة تحتوي على اسم صفات الكائن Response وإجراءاته، ومن أمثلة هذه الصفات الصفة Value التي تستخدم لإسناد قيمة داخل ملف Cookie، كما يمكن اختيار الإجراء (DateAdd) لإسناد تاريخ انتهاء الصلاحية حيث تم تحديد تاريخ انتهاء الصلاحية بعد ثلاثة أشهر من الآن باستخدام تاريخ اليوم (Today).



الشكل رقم (١٦,٧). ملفات Cookie ذات القيمة الواحدة والمتعددة القيم.

```
Private Sub btnCreateSingleCookie_Click(ByVal sender As System.Object,_
  ByVal e As System.EventArgs) Handles btnCreateSingleCookie.Click

        Response.Cookies("CustomerName").Value = "Eleanor Hankins"
        Response.Cookies("CustomerName").Expires = _
            DateAdd(DateInterval.Month, 3, Today)
End Sub

Private Sub btnDisplayCookie_Click(ByVal sender As System.Object, _
   ByVal e As System.EventArgs) Handles btnDisplayCookie.Click

    lblDisplaySingleCookie.Text = _
            Request.Cookies("CustomerName").Value
End Sub

Private Sub btnSaveMVCookie_Click(ByVal sender As System.Object,_
   ByVal e As System.EventArgs) Handles btnSaveMVCookie.Click
```

الشكل رقم (١٦,٨). شفرة لتشئ ملفات Cookies وتسترجعها.

```
        Response.Cookies("bmAddress")("StreetAddress") = "777 Sunset"
        Response.Cookies("bmAddress")("CityAddress") = "Springfield"
        Response.Cookies("bmAddress").Expires = _
            DateAdd(DateInterval.Month, 3, Today)
End Sub

Private Sub btnDisplayMVCookie_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles btnDisplayMVCookie.Click

        lblAddressCookie.Text = _
            Request.Cookies("bmAddress")("StreetAddress")
        lblCityCookie.Text = _
            Request.Cookies("bmAddress")("CityAddress")
End Sub
```

تابع الشكل رقم (١٦,٨).

كما يتم استرجاع قيمة ملف Cookie ذات القيمة الواحدة باستخدام الكائن Request وإسنادها للخاصية Text للعنوان lblDisplaySingleCookie لإظهارها على صفحة الويب وذلك عند الضغط على زر العرض.

يختلف ملف Cookie متعدد القيم عن ملف Cookie ذات القيمة الواحدة حيث يجب كتابة اسم ملف Cookie أولاً ثم اسم العنصر، فإذا عدنا إلى المثال السابق نجد أن اسم ملف Cookie متعدد القيم هو bmAddress، ويحتوي هذا الملف على عنصرين هما: العنصر StreetAddress، والعنصر CityAddress، وكما يتضح أن إجراء تخزين ملف Cookie متعدد القيم يكتب قيم العناصر داخل الملف، ثم يسند تاريخ صلاحية له، أما وظيفة عرض بيانات ملف Cookie متعدد القيم فهي استرجاع قيم عناصر الملف وإسنادها للخاصية Text للعنوانين المتواجدين على صفحة الويب.

يوضح هذا المثال استخدام كل من ملف Cookie متعدد القيم وملف Cookie ذو القيمة الواحدة، ولتجربة هذا المثال نقوم بالضغط على زر إنشاء كل من ملف Cookie متعدد القيم وملف Cookie ذو القيمة الواحدة، ثم نقوم بغلق التطبيق، ثم نقوم بتشغيله مرة أخرى والضغط على كل من زر عرض ملف Cookie متعدد القيم وملف Cookie ذو القيمة الواحدة، وستظهر لك البيانات على صفحة الويب كما هو واضح في الشكل رقم (١٦,٧).

التحكم في حالة جلسة ASP.NET
Using the ASP.NET Session State Control

تستخدم تقنية ASP.NET بشكل افتراضي ملفات Cookie لتحتفظ بطلبات المستخدم أثناء جلسة زيارته، حيث تتبع نظاماً خاصاً لمعالجة هذه الملفات (لا تستخدم تقنية ASP.NET كلاً من الكائن Response والكائن Request، كما تتواجد ملفات Cookie أثناء جلسة زيارة المستخدم فقط للموقع)، ولكن مع زيادة الوعي حول المخاطر الأمنية تم التفكير

في بديل لملفات Cookies للتحكم في جلسة المستخدم ؛ ولذلك يمكن ضبط ASP.NET لكي يعمل دون استخدام ملفات Cookies وتستخدم نموذج حالة الجلسة الخاص بها حيث يحفظ خادم ASP.NET جميع الكائنات داخل المجموعة Session داخل مخزن حالة الجلسة مع نهاية كل طلب صفحة ويب، مع العلم أن ASP.NET تحتفظ بمخزن حالة الجلسة داخل الذاكرة الفورية (Cache Memory) ولكن يمكن ضبطها لكي تحتفظ بحالة الجلسة داخل قاعدة بيانات.

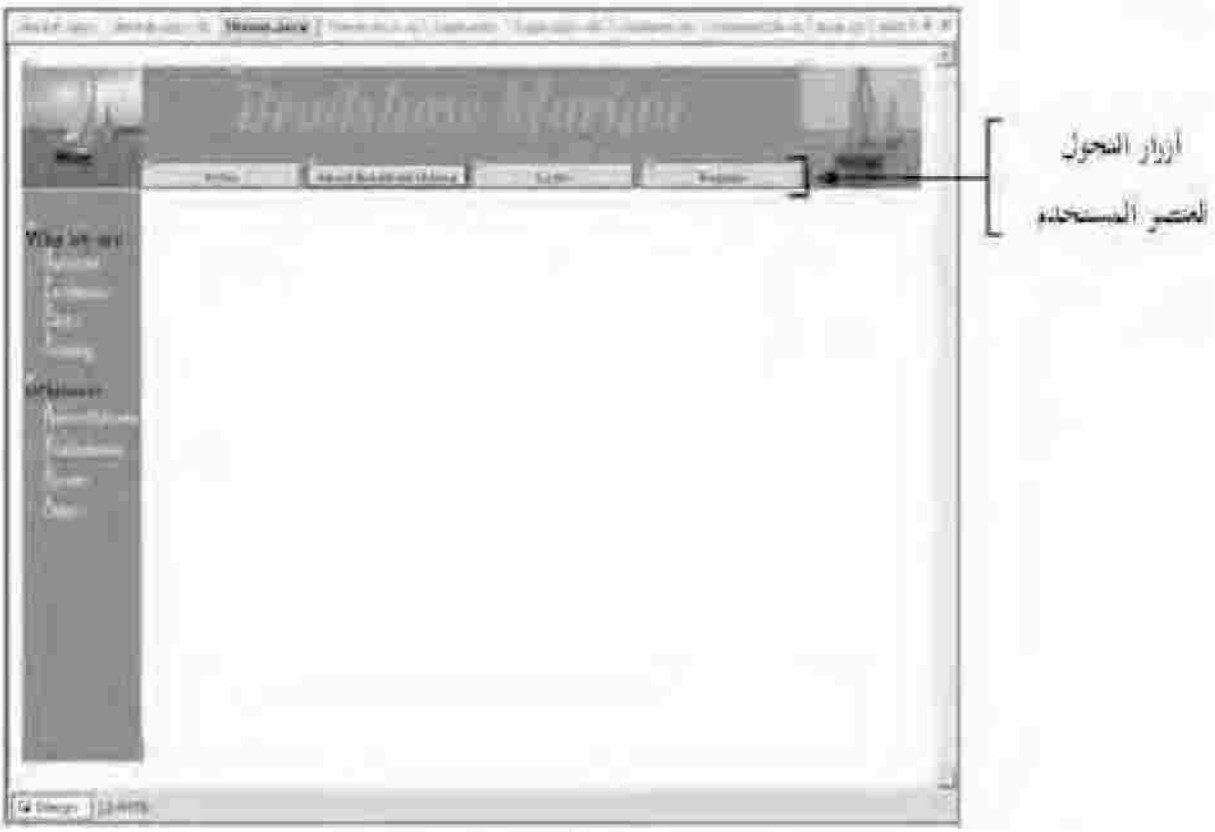
إنشاء عنصر مستخدم ASP.NET
Creating an ASP.NET User Control

إن "عنصر مستخدم" (User Control) عبارة عن نموذج ويب يتكون من عناصر خادم ASP وشفرة لغة HTML دون وجود العلامة <HTML> والعلامة <HEADER> والعلامة <TITLE>. يتم إنشاء ملف عنصر المستخدم داخل تطبيق الويب ويأخذ الامتداد "ascx.". يقدم عنصر المستخدم إمكانية تطوير أجزاء مشتركة بين عدة نماذج ويب داخل عنصر مستخدم ثم إدراج هذا العنصر داخل تلك النماذج مما يؤدي إلى تطوير الأجزاء المشتركة مرة واحدة فقط، ولكن يجب تسجيل عنصر المستخدم قبل استخدامه كما سنرى خلال هذا الفصل، فعلى سبيل المثال يمكن إنشاء وحدة تجول داخل صفحات الويب على هيئة عنصر مستخدم، ثم إدراج هذا العنصر داخل جميع صفحات الويب وذلك يقدم لنا ميزة التعديل في عنصر المستخدم فقط (أي التعديل في ملف واحد فقط)، ومن ثم سيؤثر على جميع صفحات الويب.

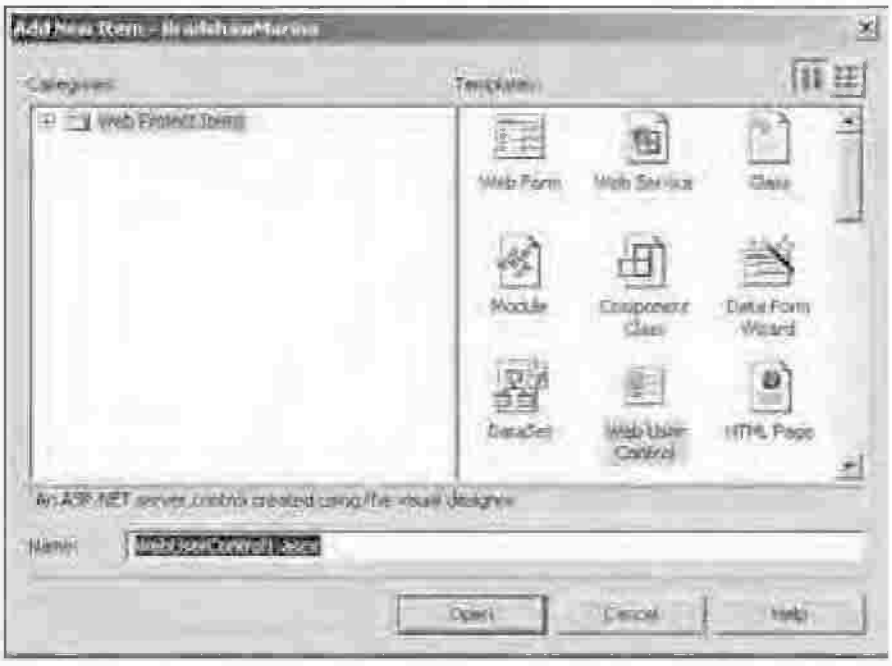
عزيزي القارئ تذكر من الفصل الثاني عشر أن عنصر المستخدم يمثل بديلاً جيداً لإطارات لغة HTML، كما أن عناصر المستخدم مألوفة لمبرمجي ASP الذين اعتادوا استخدام برامج الخادم ودمجها باستخدام أمر "Includes".

يمكن إنشاء عناصر المستخدم بسهولة باستخدام نماذج ASP.NET، فعلى سبيل المثال يشتمل موقع شركة برادشو مارينا على صف أفقي من أزرار التجول التي تستخدم للتجول بين صفحات الموقع كما هو واضح من الشكل رقم (١٦,٩)، ولأن ملف عنصر المستخدم هذا يتواجد داخل تطبيق الويب فسيُظهِر مستكشف الحلول ملفات أخرى.

يتكون الجزء العلوي من نموذج عنصر المستخدم من صورة لمركب شراعي تظهر على اليمين وعلى اليسار، كما تحتوي على عنوان شركة برادشو مارينا وتحتوي أيضاً على أزرار تجول وهي: الزر "Home"، والزر "About Bradshaw Marina"، والزر "Login"، والزر "Register". يحتوي الجزء الأيسر للنموذج على عناصر نص والتي يستخدم بعضها كعناصر تجول مثل العنصر "Service"، والعنصر "Location". ولكي تنشئ عنصر مستخدم، قم بالضغط على الزر الأيمن للفأرة على المشروع الحالي واختر الأمر "Add new item"، فعندئذ سيظهر الصندوق الحواري الواضح في الشكل رقم (١٦,١٠)، ثم اختر الرمز "Web User Control" الموضح في الجزء الأيمن للنافذة.

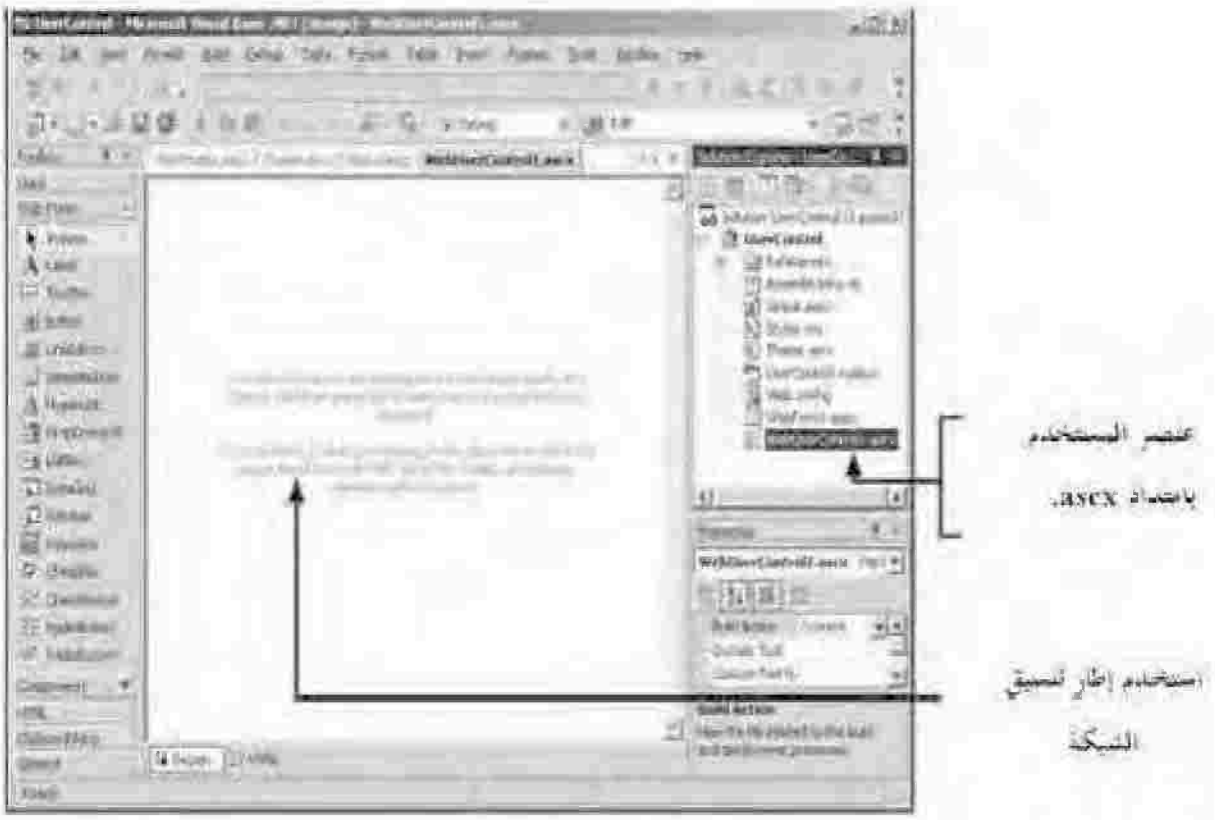


الشكل رقم (١٦,٩). عنصر المستخدم للتحول بين الصفحات.



الشكل رقم (١٦,١٠). نافذة تحتوي مكونات Controls وتسجيلها.

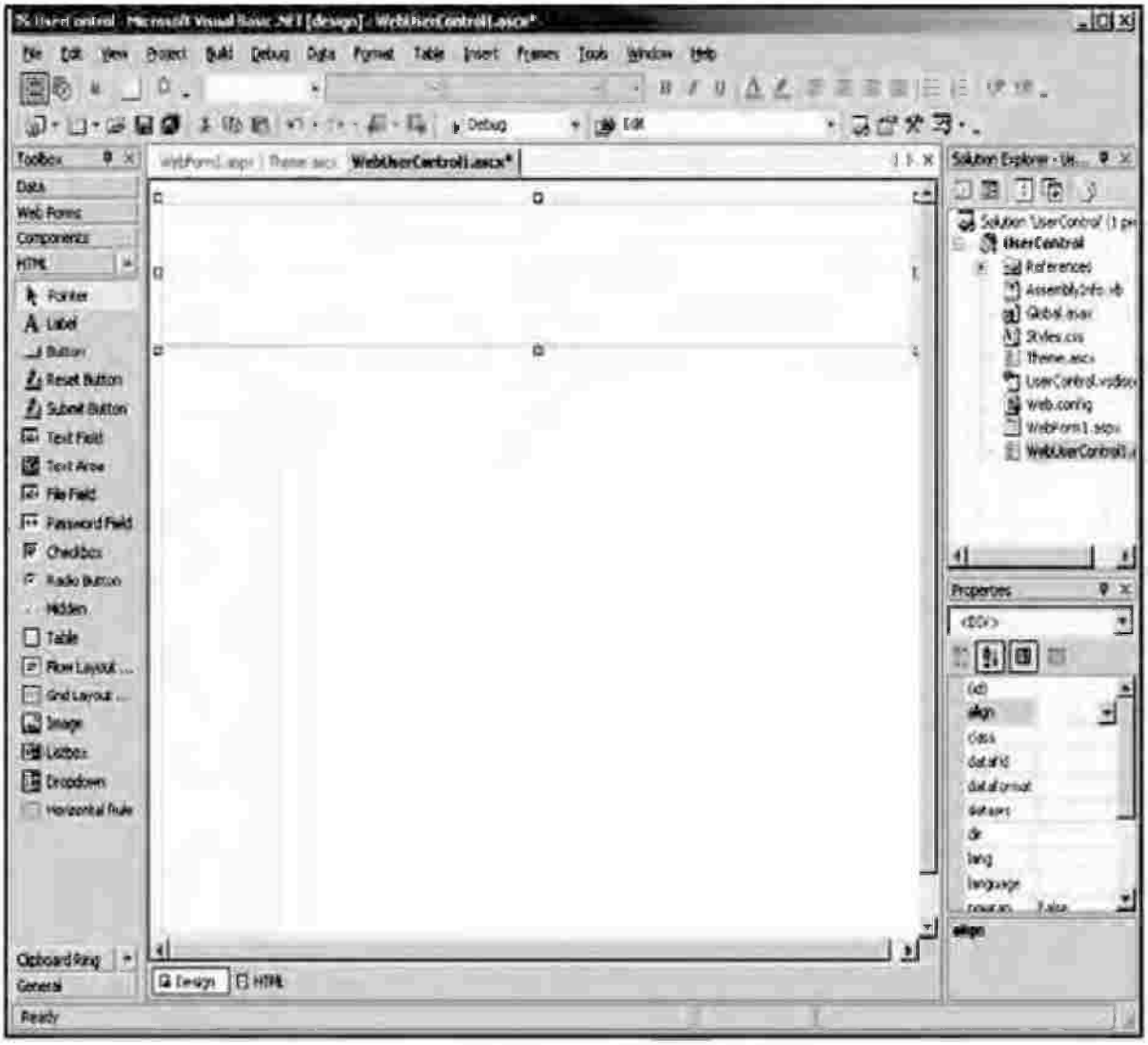
ولكي تقبل الاسم الافتراضي للعنصر، اضغط على الزر Open وعندئذ ستظهر نافذة تصميم عنصر المستخدم كما هو واضح في الشكل رقم (١٦،١١) (ربما تحتوي النافذة الظاهرة أمامك على ملفات مختلفة).



الشكل رقم (١٦،١١). تصميم نافذة تصميم التحكم.

يتطلب تنسيق العناصر داخل نموذج عنصر المستخدم استخدام الإطار "GridLayout" كما هو واضح في الشكل رقم (١٦،١٢).

وبعد وضع الإطارات (Gridlines) المطلوبة على النموذج، يتم اختيار وضع عناصر الويب على عنصر المستخدم، فعلى سبيل المثال، لتصميم الجزء العلوي للتجول الخاص بموقع شركة برادشو ماربنا، نحتاج أن نضع عنصر الصورة (Image Control) داخل الإطار Gridline العلوي حيث نحتاج وضع واحدة جهة اليسار والأخرى جهة اليمين لكي يتشابه التصميم مع الشكل رقم (١٦،٩). ولتحميل صورة المركب الشراعي يجب البحث على الخاصية ImageURL لعنصر الصورة ثم الضغط على الزر Browse لاختيار الصورة المناسبة وإسنادها لعنصر الصورة.

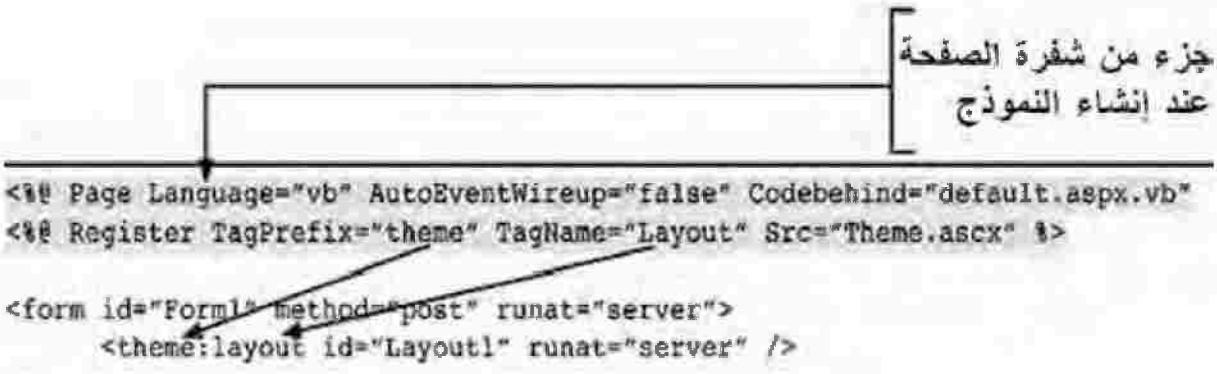


الشكل رقم (١٦,١٢). عنصر تحكم إطار GridLayout.

ولأن الأزرار وبعض عناصر النص تستخدم للتجول فنحتاج أن نختار عناصر تستجيب لحدث الضغط الزر الأيسر للفأرة؛ ولذلك يمكن استخدام عناصر خادم الويب أو استخدام عناصر HTML ذات الصفة Anchor (أي رابطة استدعاء)، وتتميز عناصر HTML بالمعالجة لدى عميل الويب عند حدوث بعض الأحداث مثل الحدث OnBlur، الحدث OnMouseOver. ومن ناحية أخرى فإن عناصر خادم الويب تعالج لدى الخادم حيث يتم إرسال آخر وضع لعناصر خادم الويب إلى الخادم عند حدوث حدث ما مثل الحدث Click، فعلى سبيل المثال إذا أردنا أن نظهر صفحة الدخول للموقع عند الضغط على زر Login فنحتاج أن نستدعي الإجراء Redirect فقط من الكائن Response داخل إجراء معالجة الحدث Click على هذا الزر كما هو واضح من الأوامر التالية:

```
Private Sub btnLogin_Click(ByVal sender As System.Object, _
   ByVal e As System.EventArgs) Handles btnLogin.Click

   Response.Redirect("Login.aspx")
End Sub
```

بعد الانتهاء من نموذج عنصر المستخدم وكتابة الأوامر المطلوبة لأزرار تجول الإطار العلوي نحتاج أن ننجز مهمتين قبل استخدام عنصر المستخدم داخل صفحات الويب. نحتاج أولاً أن نسجل نموذج الويب باستخدام علامة <Page> حيث يتم تحويل عنصر المستخدم إلى عنصر خادم ويب ASP بإضافة الصفة TagName والصفة TagPrefix، ولعمل ذلك قم بإظهار رؤية HTML للصفة الحالية، ثم أضف علامة <%@ Register> بعد علامة <% Page %> التي تنتج بشكل آلي كما هو واضح في الشكل رقم (١٦,١٣)، لاحظ أن القيمة "Theme" تم إسنادها للصفة TagPrefix والقيمة "Layout" تم إسنادها للصفة TageName.



الشكل رقم (١٦,١٣). الصفحة المباشرة.

أما المهمة الثانية فهي تهتم بإضافة عنصر المستخدم داخل نموذج الويب، ولعمل ذلك يجب إضافة علامة Prefix الخاصة بعنصر المستخدم داخل نموذج الويب كما هو واضح في الشكل رقم (١٦,١٣) (لاحظ أنه قد تم حذف الكثير من الأوامر والتركيز على الأوامر المهمة فقط)، وبعد الانتهاء من هاتين المهمتين سيتم تحميل عنصر المستخدم داخل نماذج الويب التي احتوت على شفرة عنصر المستخدم.

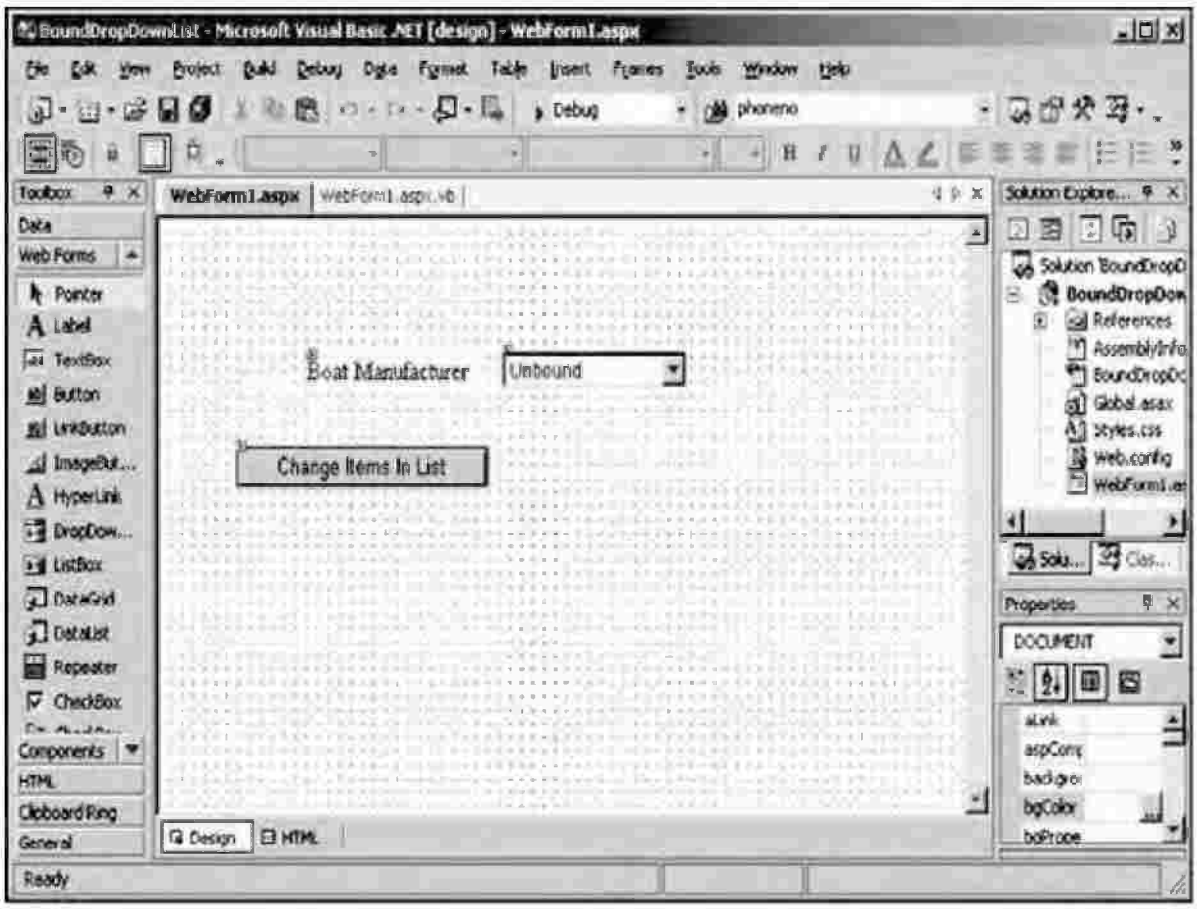
استخدام تقنية ربط البيانات Data Binding
Using Data Binding Technology

تقدم تقنية ربط البيانات (Data Binding) وسيلة سهلة لارتباط خاصية عنصر ما بمصدر للبيانات، فمثلاً يمكن ربط الخاصية Text لصندوق نص بحقل داخل جدول في قاعدة البيانات، ومن ثم ستكون قيمة صندوق النص هي

قيمة الصف الحالي لهذا العمود، ويطلق على هذه العناصر اسم عناصر مرتبطة بالبيانات (Data Bound Controls)، حيث توفر تقنية ربط البيانات بعناصر واجهة الاستخدام كتابة أوامر لإعادة إظهار البيانات بعد تغييرها داخل مصدر البيانات حيث يتم ذلك بشكل تلقائي. وعادة يكون مصدر البيانات هو جدول داخل قاعدة بيانات، ولكن يمكن أن يكون مصدر البيانات عبارة عن مجموعة بيانات مثل مصفوفة تحتوي على بيانات.

ربط البيانات بقائمة منسدلة
Binding Data to a Drop-Down List

يحتوي نموذج الويب الموضح في الشكل رقم (١٦،١٤) على ثلاثة عناصر واجهة استخدام وهي: قائمة منسدلة (Drop-Down List)، وعنوان (Label)، وزر (Button). نقوم بإسناد أسماء أربعة مصانع مراكب داخل قائمة منسدلة بواسطة قيم مخزنة داخل مصفوفة، ويعمل الزر "Change Items in List" على تغيير قيم هذه المصفوفة التي تنعكس مباشرة على القائمة المنسدلة.



الشكل رقم (١٦،١٤). تصميم نموذج الويب بثلاثة عناصر تحكم.

توضح الأوامر التالية إنشاء المصفوفة وربط قيمها بالقائمة المنسدلة:

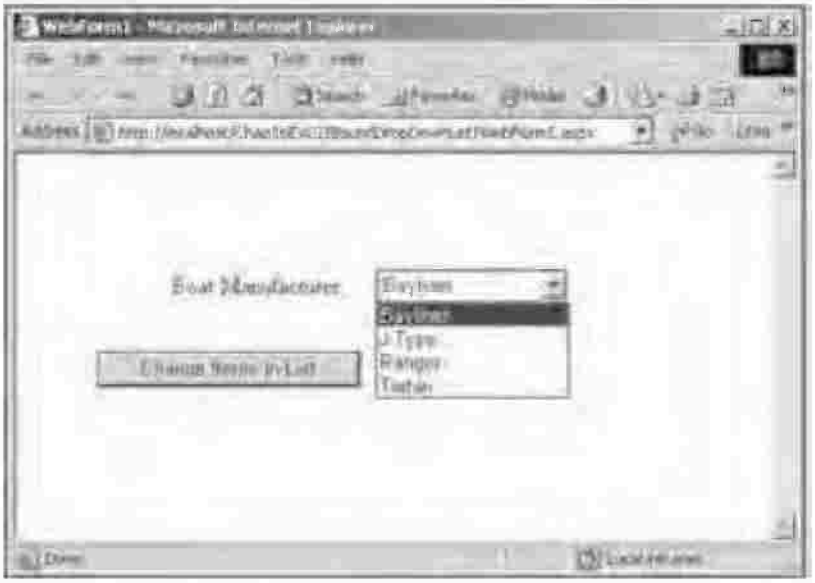
```
Dim boatManufacturers As New ArrayList()

 Private Sub bindArrayList()
     ddlBoats.DataSource = boatManufacturers
     ddlBoats.DataBind()
 End Sub

 Private Sub Page_Load(ByVal sender As System.Object, _
  ByVal e As System.EventArgs) Handles MyBase.Load
     'Put user code to initialize the page here

     boatManufacturers.Add("Ranger")
     boatManufacturers.Add("J-Type")
     boatManufacturers.Add("Tartan")
     boatManufacturers.Add("Bayliner")

     boatManufacturers.Sort()

     bindArrayList()

 End Sub
```

لاحظ أن المصفوفة boatManufactures تم تعريفها خارج جميع الإجراءات ليتم استخدامها داخلها، ثم يأتي إجراء Page_Load المسؤول عن إضافة أربعة أسماء لمصانع مراكب داخل المصفوفة، ثم فرز هذه البيانات بواسطة استدعاء الإجراء Sort، ثم استدعاء الإجراء BindArrayList المسؤول عن ربط المصفوفة بالقائمة المنسدلة. يسند هذا الإجراء اسم المصفوفة boatManufactures إلى الخاصية DataSource للقائمة المنسدلة، ثم يستدعي الإجراء DataBind من القائمة المنسدلة لإتمام عملية ربط البيانات.

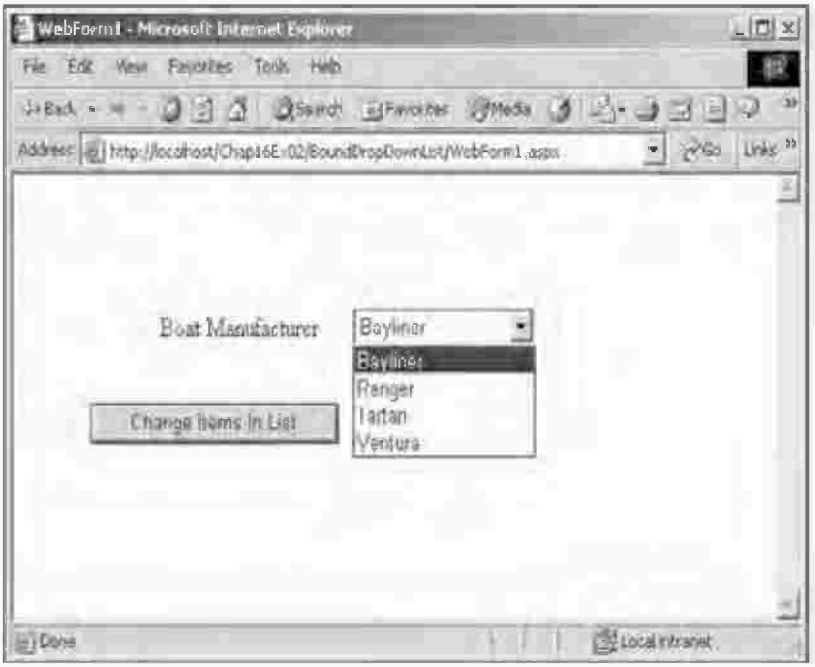
كما أن الصنف Page يحتوي على الإجراء DataBind الذي إذا تم استدعاؤه (Page.DataBind) فسيربط جميع عناصر النموذج الحالي بمصادر البيانات المرتبطة بها. توضح الأوامر التالية إجراء معالجة زر تغيير البيانات داخل القائمة حيث يتم حذف قيمة من المصفوفة، ثم إضافة قيمة أخرى، ثم استدعاء الإجراء BindArrayList لتجديد ربط القائمة المنسدلة بالمصفوفة.

```
Private Sub btnChangeValue_Click(ByVal sender As System.Object, _
     ByVal e As System.EventArgs) Handles btnChangeValue.Click

     ' Change the values in the ArrayList
     boatManufacturers.Remove("J-Type")
     boatManufacturers.Add("Ventura")

     ' Invoke the method to bind the ArrayList values
     bindArrayList()

End Sub
```

يوضح الشكل رقم (١٦,١٥) صفحة ويب مصانع المراكب قبل تغيير محتوى القائمة المنسدلة (أي قبل تغيير محتوى المصفوفة)، كما يوضح الشكل رقم (١٦,١٦) صفحة الويب نفسها ولكن بعد تغيير محتوى القائمة والمصفوفة، ولاحظ أن القيمة "J-Type" تم حذفها واستبدالها بالقيمة "Ventura"، وأن القائمة المنسدلة تظهر بشكل مفهرس.



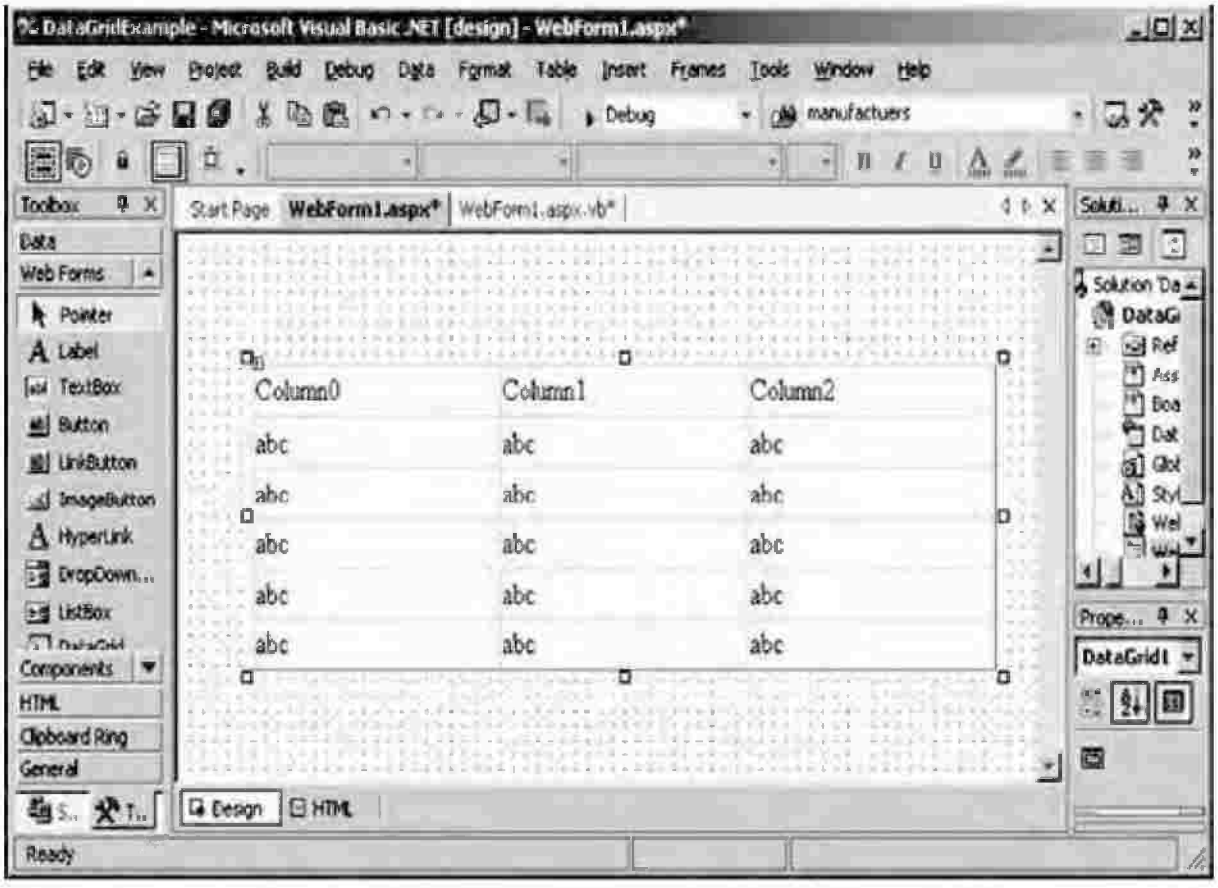
الشكل رقم (١٦,١٥). قائمة منسدلة تعرض قيم مصفوفة مبدئية.



الشكل رقم (١٦,١٦). قائمة منسدلة تعرض بعد تغيير قيمة المصفوفة.

التعرف على عنصر واجهة الاستخدام DataGrid
Understanding the DataGrid Control

تقدم تقنية ASP.NET ثلاثة عناصر يمكن ربطها بالبيانات والتي تم تصميمها لعرض البيانات على نماذج الويب وهي: العنصر Repeater، والعنصر DataList، والعنصر DataGrid. يشبه العنصر DataGrid جدول HTML ولكن مع وجود مميزات جيدة مثل فهرسة أعمدة الجدول، وتقسيم محتوى الجدول على عدة صفحات (Pagination)، وتعديل البيانات داخل الجدول، وتنسيق البيانات بشكل مرن. ولأن عنصر DataGrid يعد أغنى العناصر الثلاثة في كافة الإمكانات فسوف يتم التركيز عليه خلال هذا الفصل. سوف نستخدم العنصر DataGrid للتعامل مع بيانات المراكب (كائنات الصنف Boat) داخل تطبيق برادشو مارينا؛ ولذلك يجب إضافة الصنف Boat من الفصل التاسع (ولإضافة صنف إلى التطبيق الحالي، اختر القائمة File، ثم اختر الأمر Add Existing Item)، ولكن تذكر من الفصل التاسع أنه توجد علاقة ترابط بين الصنف Boat والصنف Customer؛ ولذلك نحتاج أيضاً إلى إضافة الصنف Customer إلى التطبيق الحالي، ثم نقوم بوضع عنصر DataGrid على نموذج الويب وتنسيقه ليظهر كما هو واضح في الشكل رقم (١٦,١٧).



الشكل رقم (١٦,١٧). عنصر DataGrid لنموذج ويب.

يمكنك إضافة المصفوفة Boats (كائن من الصنف ArryList) ثم جعله مصدر بيانات للعنصر DataGrid المضاف إلى نموذج الشكل رقم (١٦,١٧)، ولإضافة مراكب (كائنات الصنف Boat) إلى المصفوفة Boats اكتب الأوامر التالية داخل إجراء معالجة الحدث Page_Init الذي سوف ينفذ عند أول طلب للصفحة، ولاحظ أن الأوامر تفترض أن المصفوفة Boats معرفة كمتغير عام، ولاحظ أيضاً أنه يجب توسعة نافذة مصمم النماذج لكي نتمكن من مشاهدة إجراء معالجة الحدث Page_Init، ثم أدخل الأوامر التالية بعد الإجراء InitializeComponent().

```
Private Sub Page_Init(ByVal sender As System.Object, _
     ByVal e As System.EventArgs) Handles MyBase.Init
     'CODEGEN: This method call is required by the Web Form Designer
     'Do not modify it using the code editor.
     InitializeComponent()

     boats.Clear()
     boats.Add(New Boat("MO12345", 20, "Ranger", 1996))
     boats.Add(New Boat("MO23456", 24, "Bayliner", 1998))
     boats.Add(New Boat("MO34567", 26, "Ventura", 2003))
     boats.Add(New Boat("MO45678", 22, "Bayliner", 1999))
     boats.Add(New Boat("MO56789", 30, "Bayliner", 2004))
     boats.Add(New Boat("MO11112", 22, "Ventura", 1996))
     boats.Add(New Boat("MO21111", 27, "Ranger", 1997))
     boats.Add(New Boat("MO12121", 24, "Bayliner", 2000))
     boats.Add(New Boat("MO22222", 33, "Ranger", 2001))
     boats.Add(New Boat("MO33123", 20, "Ranger", 2003))
     boats.Sort(0, 0, Nothing)
 End Sub
```

ترتبط عناصر DataGrid مع قواعد البيانات في التطبيقات التي تتكون من طبقتين (Two-Tier)، أما التطبيقات التي تتكون من ثلاث طبقات (Three-Tier) فإن المصفوفات تتوسط كلاً من عناصر واجهة الاستخدام وقواعد البيانات. يستخدم الصنف DataTable لمعالجة البيانات والربط مع عناصر واجهة الاستخدام، ولإنشاء جدول (كائن من الصنف DataTable)، يجب إنشاء أعمدة (Columns) وصفوف (Rows) كما هو واضح من الشكل رقم (١٦,١٨) حيث توجد أوامر الإجراء المسؤول عن إنشاء جدول (الكائن dt) واستخلاص القيم من المصفوفة Boats، ثم إضافتها إلى هذا الجدول.

ولإنشاء هذا الجدول، يجب عليك أولاً إنشاء الكائن dt من الصنف DataTable والكائن dr من الصنف DataRow، ثم أضف أربعة أعمدة إلى المجموعة Collections للكائن dt باستخدام أسماء الأعمدة المناسبة والمناظرة لصفات الصنف Boat، وتوضح الأوامر التالية إنشاء هذه الأعمدة.

```
Dim dt As New DataTable()
Dim dr As DataRow
dt.Columns.Add(New DataColumn("StateRegNo", GetType(String)))
dt.Columns.Add(New DataColumn("BoatLength", GetType(Integer)))
dt.Columns.Add(New DataColumn("Manufacturer", GetType(String)))
dt.Columns.Add(New DataColumn("Year", GetType(Integer)))
```

وبعد إنشاء الجدول (الكائن dt) نحتاج أن نقوم باستخلاص محتويات المصفوفة Boats وإسنادها إلى الجدول dt؛ ولذلك يجب تعريف متغير إشارة MyBoat من النوع Boat للتجول داخل المصفوفة Boats واسترجاع بيانات المراكب وإسنادها داخل صفوف هذا الجدول حيث يتم استخلاص بيانات الكائن MyBoat في كل مرة تجول وإسنادها إلى أحد صفوف الجدول، ثم إضافة هذا الصف داخل الجدول كما هو واضح من الأوامر التالية:

```
For Each myboat In boats
    dr = dt.NewRow

    dr("stateRegNo") = myboat.GetStateRegistrationNo
    dr("BoatLength") = myboat.GetLength
    dr("Manufacturer") = myboat.GetManufacturer
    dr("Year") = myboat.GetYear

    dt.Rows.Add(dr)
Next
```

والآن يمكن ربط عنصر DataGrid بالجدول dt كمصدر للبيانات، ولعمل ذلك يمكن إسناد الصفة DefaultView للجدول dt إلى متغير الإشارة dv الذي سيتم إسناده كمصدر بيانات للعنصر DataGrid مما يؤدي إلى إجراء كل من عملية فرز الجدول وفهرسة الجدول بسهولة (سوف يتم التعرض لفرز البيانات داخل عنصر DataGrid وتقسيمها إلى عدة صفحات لاحقاً)، ثم يتم استدعاء الإجراء DataBind من عنصر DataGrid لإنهاء عملية الربط.

```
' Assign a dataview to the default view of the data table
 dv = dt.DefaultView

' Sort the dataview
dv.Sort = sortkey
' Set the datagrid page
DataGrid1.CurrentPageIndex = newpage

' Set datagrid data source and bind the datagrid
DataGrid1.DataSource = dv
DataGrid1.DataBind()
```

```
' Declare an array list for holding boats
Private boats As New ArrayList()

' Declare a data view, variables for page no. and sort key
Private dv As DataView
Shared NP As Integer
Shared SK As String

Private Sub createBoatTable(ByVal boats As ArrayList, _
    Optional ByVal sortkey As String = "StateRegNo", _
    Optional ByVal newpage As Integer = 0)

    ' Define a reference variable of type Boat
    Dim myboat As Boat

    ' Define the columns in the table
    Dim dt As New DataTable()
    Dim dr As DataRow
    dt.Columns.Add(New DataColumn("StateRegNo", GetType(String)))
    dt.Columns.Add(New DataColumn("BoatLength", GetType(Integer)))
    dt.Columns.Add(New DataColumn("Manufacturer", GetType(String)))
    dt.Columns.Add(New DataColumn("Year", GetType(Integer)))

    ' Populate the data table
    For Each myboat In boats
        dr = dt.NewRow  ' create row buffer to hold data

        dr("stateRegNo") = myboat.GetStateRegistrationNo
        dr("BoatLength") = myboat.GetLength
        dr("Manufacturer") = myboat.GetManufacturer
        dr("Year") = myboat.GetYear

        dt.Rows.Add(dr) ' add the data to the data table
    Next

    ' Make sure the sortkey and page number do not create an error
    If sortkey = Nothing Then sortkey = "stateregno"
    If newpage < 0 Or newpage > DataGrid1.PageCount - 1 Then newpage = 0

    ' Assign a dataview to the default view of the data table
    dv = dt.DefaultView

    ' Sort the dataview
    dv.Sort = sortkey
    ' Set the data grid page
    DataGrid1.CurrentPageIndex = newpage
    ' Set datagrid data source and bind the datagrid
    DataGrid1.DataSource = dv
    DataGrid1.DataBind()
End Sub
```

إنشاء الأعمدة

إسناد قيم للأعمدة

الشكل رقم (١٦,١٨). إجراء لإنشاء جدول وتأهيله مع بيانات المركب.

والآن بعد إنشاء كل من الجدول dt وعنصر DataGrid وإنشاء علاقة ربط بينهما سوف نركز على ضبط خصائص وتشكيل عنصر DataGrid مثل فرز البيانات داخل DataGrid وضبط خصائص الأعمدة، وضبط خصائص عرض البيانات داخل عدة صفحات، وأخيراً تشكيل كل من الحدود الداخلية للجدول والحدود الخارجية.

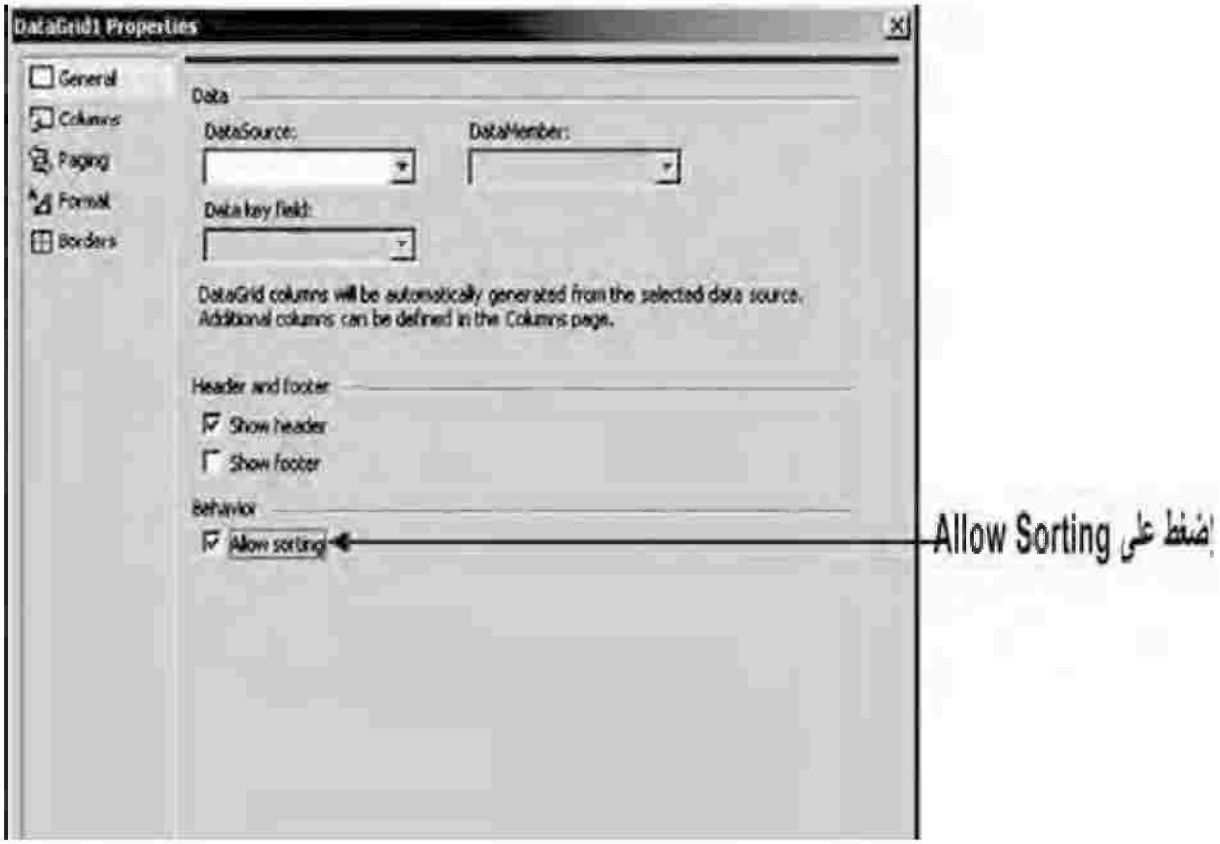
عندما نضبط عنصر DataGrid لكي يسمح بفرز البيانات داخله، يتم إرسال اسم العمود الذي سيتم الضغط عليه من قبل المستخدم بغرض فرز البيانات كمعامل إلى إجراء معالجة الضغط على رأس عنصر DataGrid ومن ثم يمكن استخلاص اسم هذا العمود (e.SortExpression) وتمريره إلى الإجراء createBoatTable لإعادة إنشاء الجدول وإعادة الربط بينه وبين عنصر DataGrid كما هو واضح من الأوامر التالية:

```
Private Sub DataGrid1_SortCommand(ByVal source As Object, _
   ByVal e As System.Web.UI.WebControls.DataGridSortCommandEventArgs) _
   Handles DataGrid1.SortCommand

   SK = e.SortExpression
   createBoatTable(boats, SK, 0)

End Sub
```

يتطلب تقسيم البيانات داخل عنصر DataGrid إجراء معالجة لتحديد فهرس الصفحة الحالية عند الضغط على رقم الصفحة (أو الضغط على الزر Next أو الزر Previous)، وبالمثل يتم استخلاص رقم الصفحة التالية (e.NewPageIndex) وتمريرها إلى الإجراء CreateBoatTable لإنشاء جدول وربطه بعنصر DataGrid كما هو واضح من الأوامر التالية:

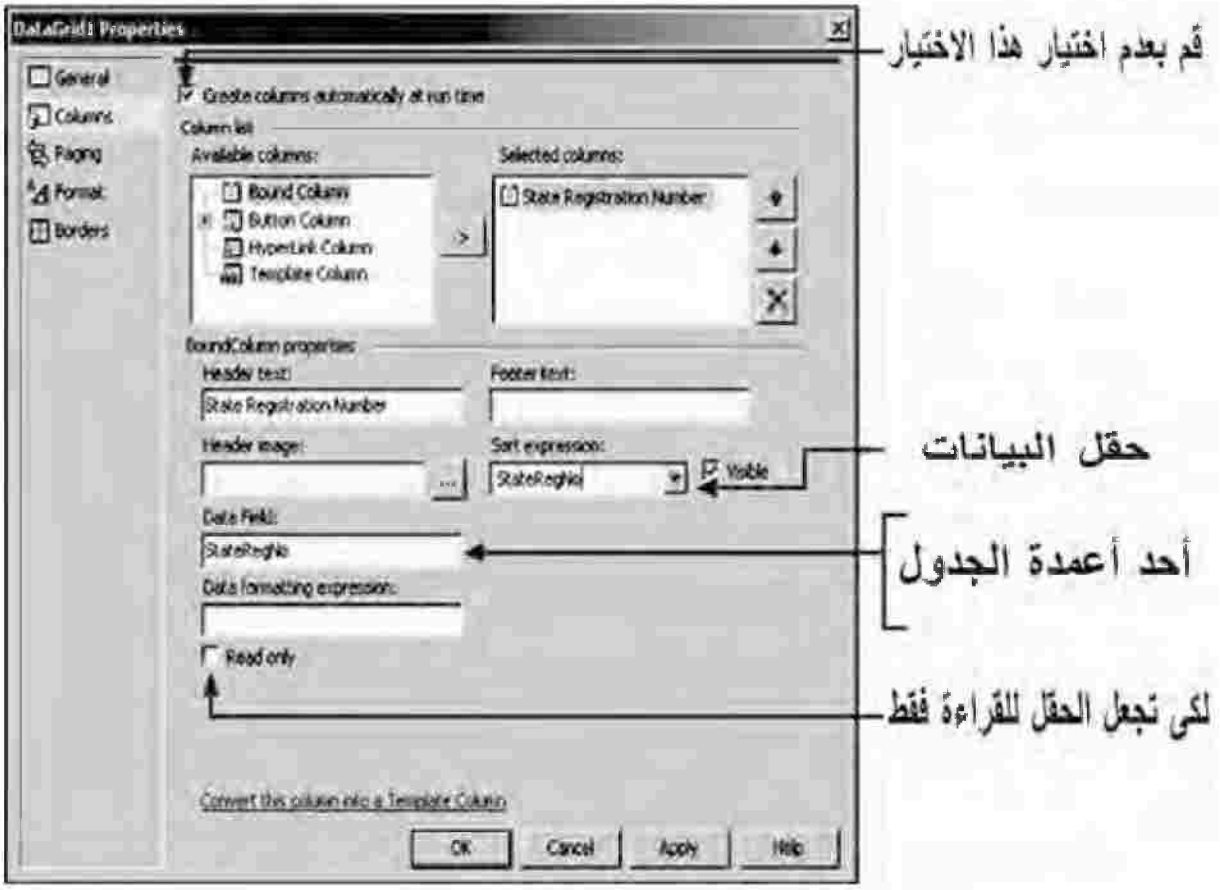
```
Private Sub DataGrid1_PageIndexChanged(ByVal source As _ Object, _
   ByVal e As System.Web.UI.WebControls.DataGridPageChangedEventArgs) _
   Handles DataGrid1.PageIndexChanged

   NP = e.NewPageIndex
   createBoatTable(boats, SK, NP)

End Sub
```

ولضبط خصائص عنصر DataGrid قم باختيار العنصر، ثم اضغط على الزر Property Builder المتواجد أسفل نافذة الخصائص لفتح نافذة الشكل رقم (١٦.١٩) والتي تتمكن من خلالها من ضبط خصائص DataGrid، حيث يمكن أن تنتقل من مجموعة خصائص إلى مجموعة أخرى بواسطة الضغط على خيارات الإطار الأيسر للنافذة التي تظهر المجموعة General عند البداية، ولفرز بيانات عنصر DataGrid قم باختيار صندوق التأشير Allow Sorting.



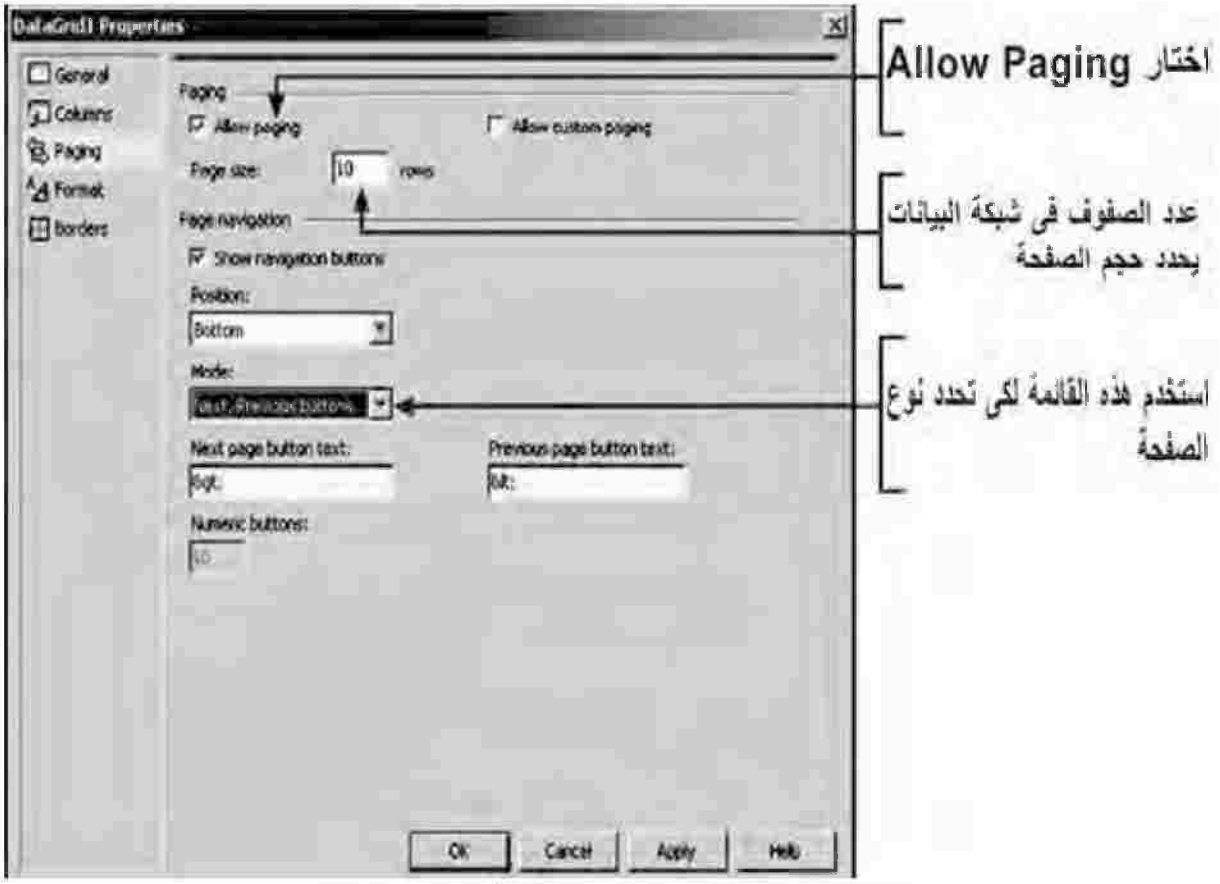
الشكل رقم (١٦,١٩). صندوق حواري لضبط خصائص عنصر DataGrid.

ولضبط خصائص الأعمدة، قم باختيار الخيار Columns من الإطار الأيسر لإظهار خصائص الأعمدة بما فيها مصطلح معيار الفرز (Sort expression) كما هو واضح في الشكل رقم (١٦.٢٠)، ولإضافة عمود كمعيار فرز البيانات قم بتحريك هذا العمود من قائمة "Available Columns" إلى قائمة "Selected Columns"، ثم قم بإدخال مصطلح معيار الفرز الذي هو عادة ما يكون اسم حقل الجدول الذي سيتم فرز البيانات على أساسه، وكرر هذه العملية لجميع الحقول التي تريد أن تشملها داخل الجدول، ولا تنسى أيضاً أن تلغي اختيار صندوق التأشير "Create columns automatically at run time" لأن الخاصية AutoGenerateColumn تأخذ القيمة True بشكل افتراضي؛ ولذلك سيقوم ASP.NET بإضافة الأعمدة المتواجدة داخل الجدول إلى عنصر DataGrid (أربع حقول في هذه الحالة)، كما يجب أن تلاحظ أن الخيار Read Only (جعل العمود للقراءة فقط) سوف يتم استخدامه لاحقاً في موقع شركة برادشو مارينا.



الشكل رقم (١٦,٢٠). تعريف عمود DataGrid.

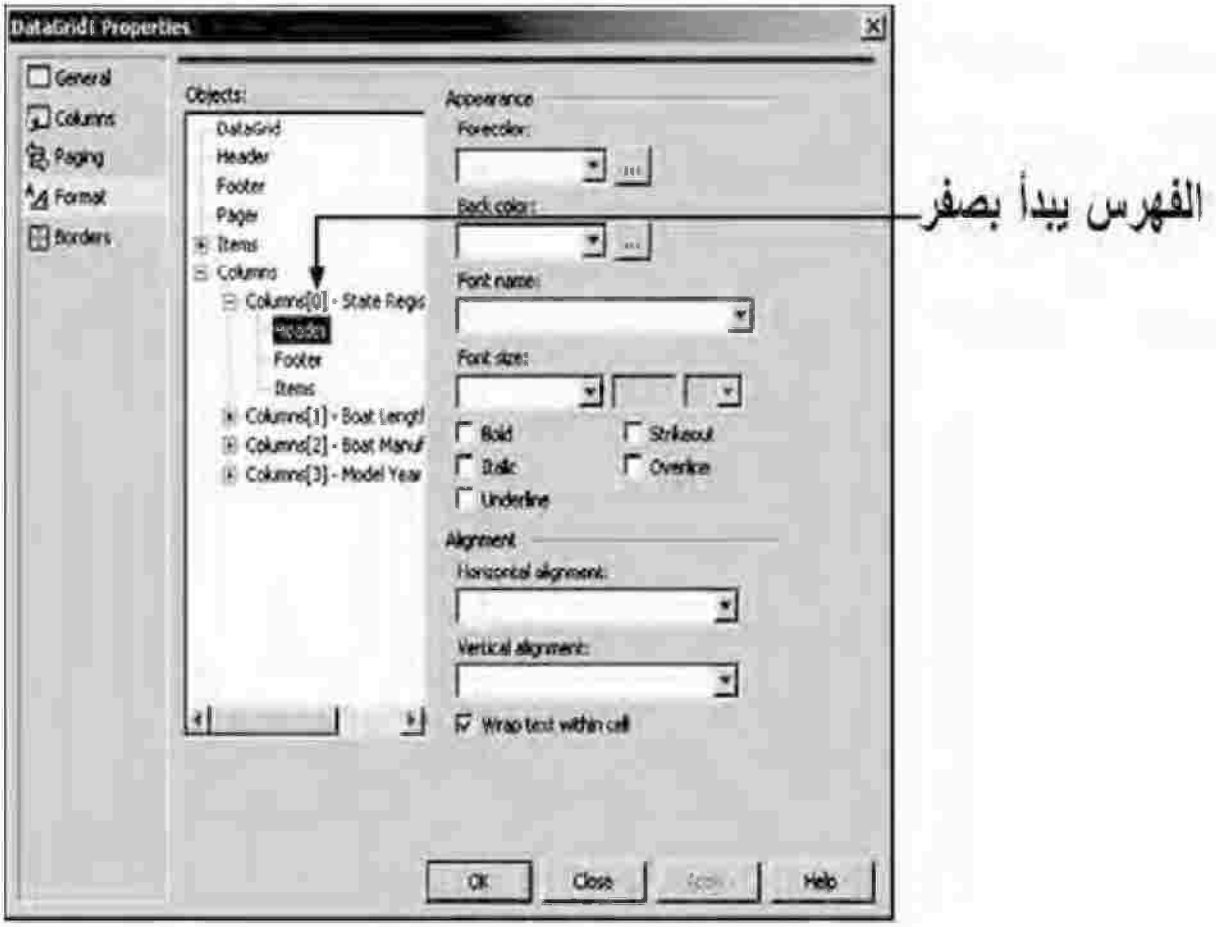
يظهر الشكل رقم (١٦,٢١) مجموعة خصائص Paging التي تحتوي على اختيار دعم ظهور البيانات داخل عدة صفحات متتالية بدلاً من ظهورها داخل صفحة طويلة، ويسمح صندوق التأشير Allow Paging بظهور البيانات داخل عدة صفحات متتالية، ويسمح لك صندوق النص Page size بتحديد صفوف الصفحة الواحدة، ولتجربة هذه الإمكانية قم بتغيير عدد صفوف الصفحة بحيث تكون أقل من عدد صفوف الجدول، فعلى سبيل المثال إذا كان عدد صفوف الجدول ١٠ صفوف فقم بتغيير عدد صفوف الصفحة إلى ٥ صفوف؛ ولذلك سوف يظهر نموذج الويب صفحتين.



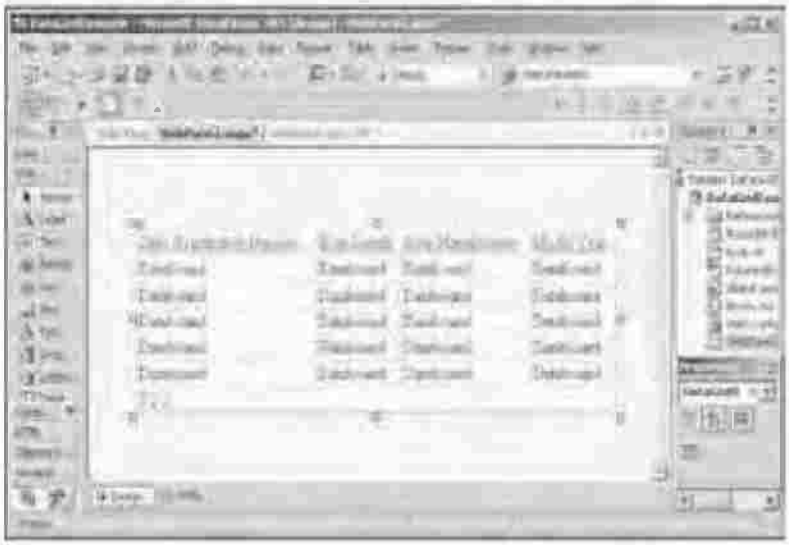
الشكل رقم (١٦,٢١). صندوق حواري لضبط خصائص Paging.

كما يمكنك اختيار الخيار Format من الإطار الأيسر كما هو واضح من الشكل رقم (١٦,٢٢) لتنسيق محتوى عنصر DataGrid وتشكيله مثل رؤوس الأعمدة وذيل عنصر DataGrid، حيث نلاحظ أن عناصر DataGrid تظهر في شكل شجري ليسهل عملية الوصول والتعامل مع العناصر، ونلاحظ أيضاً أن ترتيب الأعمدة داخل عنصر DataGrid يعتمد على فهرس يبدأ من الرقم صفر Zero-based reference.

وبعد ضبط جميع خصائص عنصر DataGrid قم بالضغط على الزر OK لنرى واجهة عنصر DataGrid في الشكل النهائي كما هو واضح في الشكل رقم (١٦,٢٣).



الشكل رقم (١٦,٢٢). صندوق حواري لضبط خصائص Format.



الشكل رقم (١٦,٢٣). صندوق حواري يظهر واجهة عنصر DataGrid.

عند بداية صفحة الويب، سيتم استدعاء الإجراء CreateBoatTable لنقل البيانات من المصفوفة إلى الجدول، ثم إجراء عملية ربط البيانات بعنصر DataGrid، ولعمل ذلك يجب استدعاء الأوامر التالية مرة واحدة وذلك عند حدوث حدث تحميل الصفحة وفحص قيمة الخاصية Postback (للتأكد من تنفيذ الأوامر عند تحميل الصفحة للمرة الأولى فقط)، ولاحظ أن الإجراء CreateBoatTable يستقبل معاملين اختياريين، ولقد تم إرسال معامل اختياري واحد في هذا الاستدعاء حيث تم استخدام رمز الفاصلة (,) وترك المعامل الثالث فارغاً ليتم استخدام القيمة الافتراضية المعرفة داخل تعريف الإجراء.

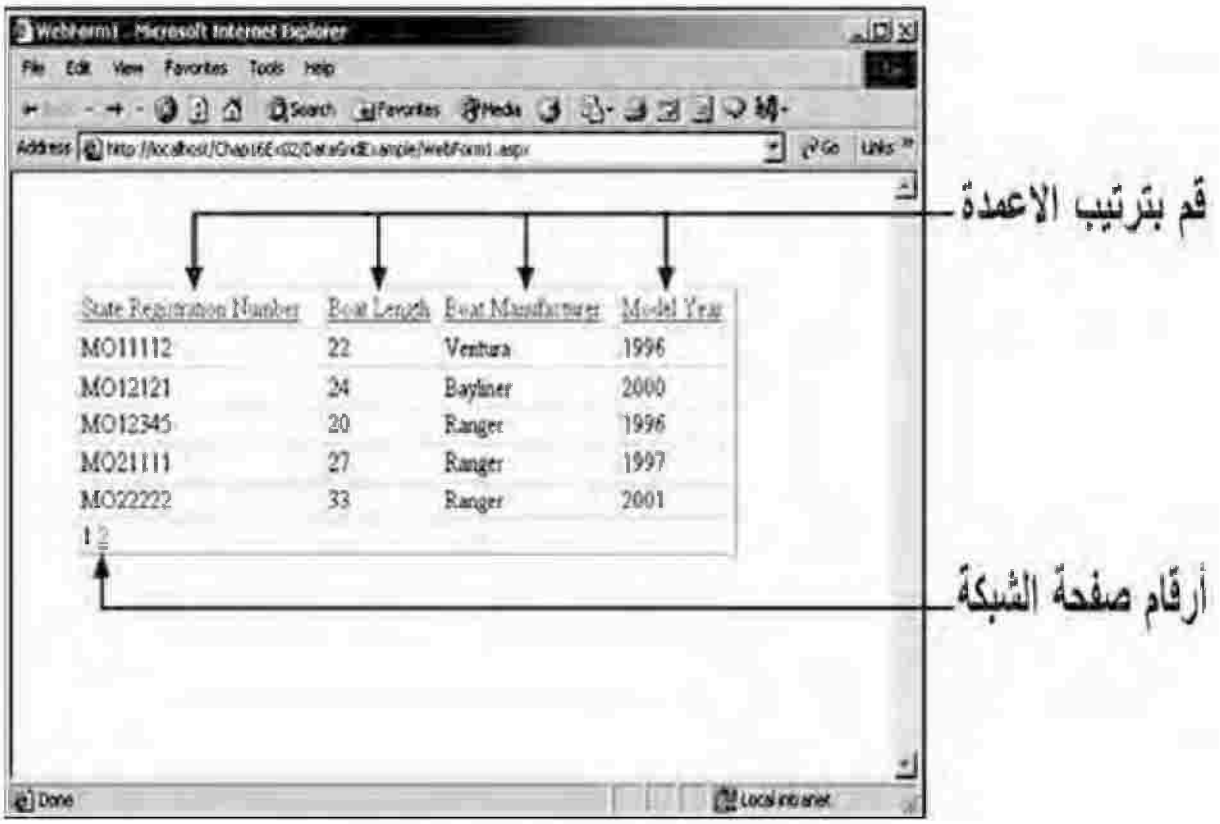
```
Private Sub Page_Load(ByVal sender As System.Object, _
   ByVal e As System.EventArgs) Handles MyBase.Load
   'Put user code to initialize the page here

   If Not IsPostBack Then
      'Invoke the method to bind boat attributes
      createBoatTable(boats, "StateRegNo", )

   End If
End Sub
```

وعند تنفيذ صفحة الويب للمرة الأولى ستظهر لك صفحة شبيهة بالصفحة الموضحة في الشكل رقم (١٦,٢٤)، ولاحظ وجود خط أسفل أسماء الحقول مما يشير إلى أن هذه الحقول يمكن الضغط عليها مما يؤدي إلى فرز البيانات الحالية طبقاً لهذا الحقل، أي يمكن ترتيب البيانات الحالية طبقاً للحقول الأربعة، ولاحظ أيضاً فهرس أرقام صفحات الموجود في الركن الأيسر لعنصر DataGrid حيث يمكن الضغط على هذه الأرقام مما يؤدي إلى تحميل بيانات الصفحة المناظرة.

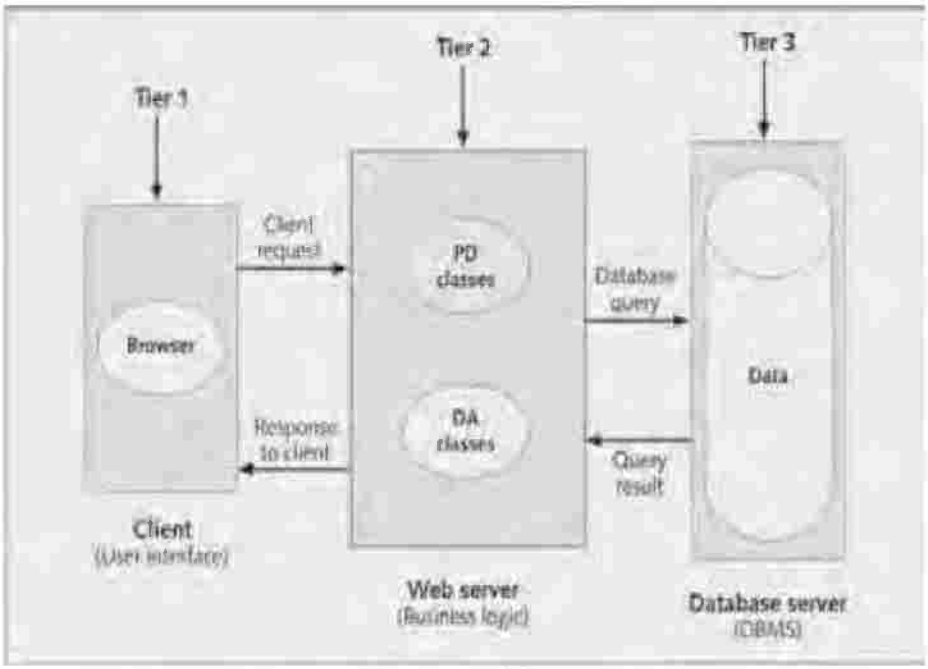
تطوير تطبيق ويب لشركة برادشو مارينا

Developing a Web Application for Bradshaw Marina

سوف نطور خلال هذه الفقرة تطبيق ويب لشركة برادشو مارينا الذي يمكن أن نعتبره نموذجاً لبناء تطبيقات ويب أخرى، مع العلم أن هذا التطبيق غير متكامل. سوف نستخدم تقنية ثلاثي الطبقات لإنشاء موقع الويب. يوضح الشكل رقم (١٦,٢٥) البنية الهيكلية لتطبيقات الويب التي تعتمد على تقنية ثلاث طبقات من الأصناف، وتذكر من الفصل الثاني عشر (وأجزاء من هذا الفصل) أن نموذج الويب الذي يعتمد على الكائن Request والكائن Response يطبق أيضاً تقنية ثلاث طبقات من الأصناف.



الشكل رقم (١٦,٢٤). واجهة تنفيذ صفحة الويب للمرة الأولى.

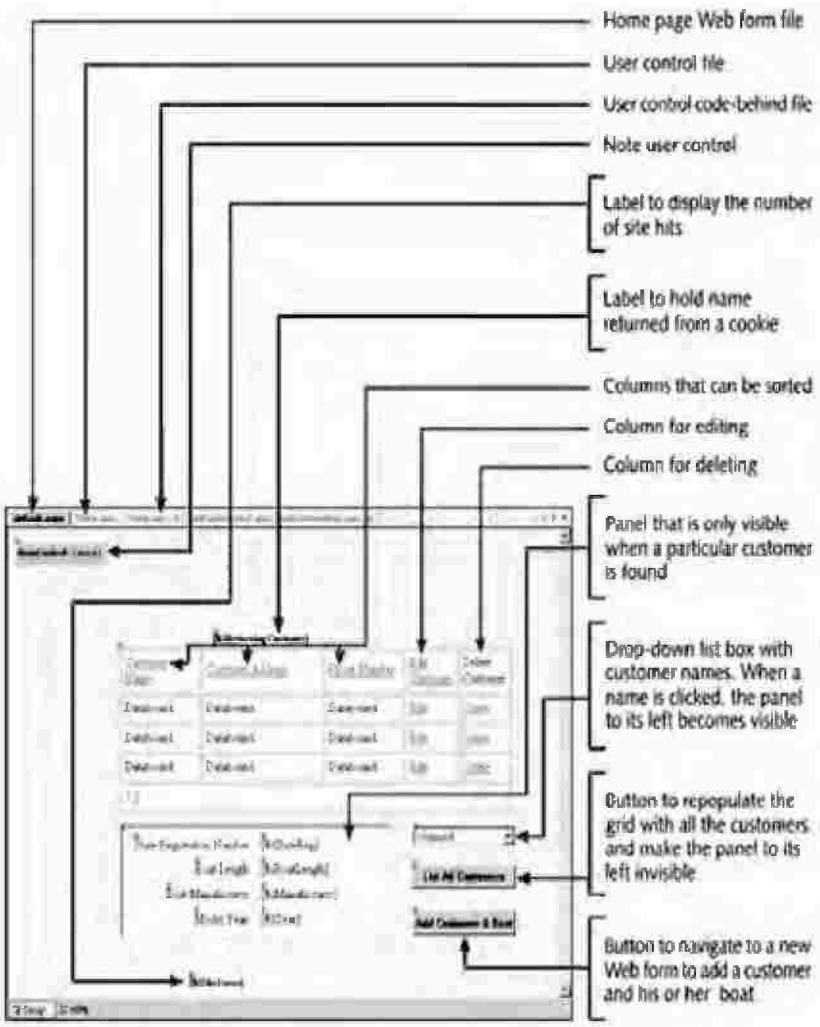


الشكل رقم (١٦,٢٥). البنية الهيكلية لتطبيقات الويب التي تعتمد على تقنية ثلاثي الطبقات.

تصميم موقع شركة برادشو مارينا
Bradshaw Marina Web Site Design

يحتوي موقع شركة برادشو مارينا على العنصر المستخدم (User Control) الذي يستخدم للتجول بين صفحات الموقع باستخدام أزرار عناصر خادم الويب، حيث يجب دمج هذا العنصر داخل جميع صفحات الموقع، ويتشابه هذا العنصر مع عنصر User Control الذي تم تطويره سابقاً داخل هذا الفصل.

يوضح الشكل رقم (١٦.٢٦) الصفحة الرئيسة للموقع والتي تحتوي على معظم العناصر المطلوبة للمعالجة داخل الموقع. يأخذ ملف الصفحة الرئيسة الاسم Default.aspx بينما يأخذ ملف عنصر User Control الاسم Theme.ascx، ويأخذ الملف الذي يحتوي على الأوامر الخاصة به الاسم Theme.ascx.vb، كما تشير صفحة الويب الرئيسة أن عنصر المستخدم قد تم تسجيله.



الشكل رقم (١٦.٢٦). الصفحة الرئيسة لموقع برادشو مارينا.

تصميم الصفحة الرئيسة لموقع شركة برادشو مارينا
Designing the Bradshaw Marina Home Page

كما هو واضح من الشكل رقم (١٦,٢٦)، فإن الصفحة الرئيسة تحتوي على عنصر DataGrid الذي تم ربطه ببيانات العملاء (Customers): الاسم (Name)، والعنوان (Address)، ورقم الهاتف (Phone Number) بالطرق التي تم شرحها في الفقرة السابقة، كما يوجد هناك عمودين يمكّنان المستخدم من تعديل بيانات العميل وحذفها (سنوضح تفاصيل التعديل والحذف لاحقاً).

عندما ينجح المستخدم في إدخال كلمة سر واسم صحيح، يتم تخزين ملف Cookie الذي يحتوي على كل من اسم العميل وعنوانه، وعندما يقوم المستخدم بزيارة الموقع مرة ثانية فسيتم إظهار اسم العميل باستخدام العنوان lblReturningCustomer المتواجد أعلى عنصر DataGrid وذلك في حال عدم انتهاء فترة صلاحية ملف Cookie.

يرتبط صندوق القائمة المنسدلة المتواجد أسفل عنصر DataGrid بأسماء العملاء، وعندما يختار المستخدم أحد أسماء هذه القائمة يتم تنفيذ الأحداث التالية:

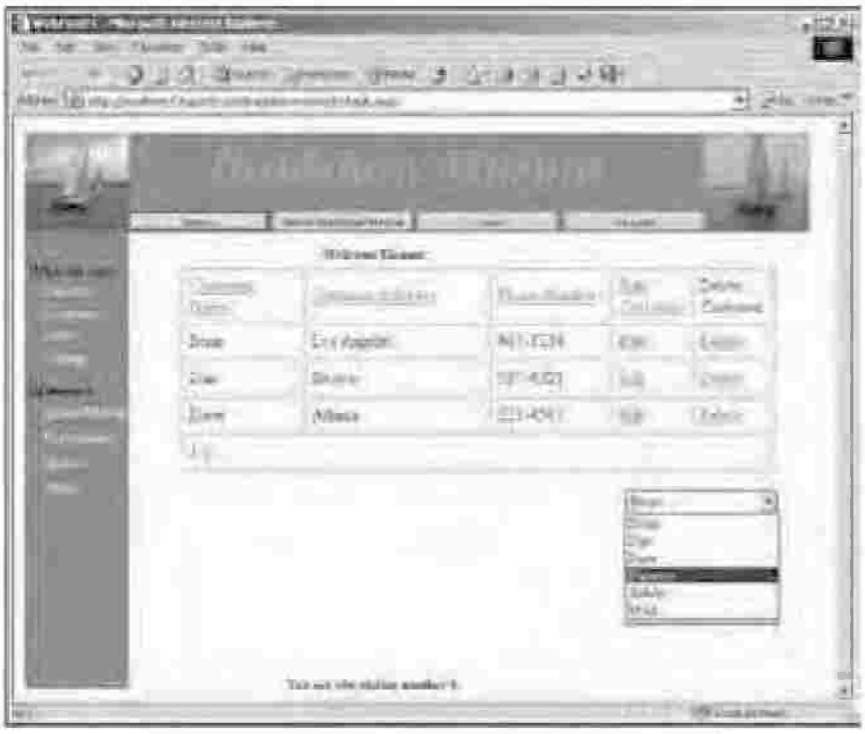
- يتم استرجاع بيانات المركب الذي يملكه العميل المختار.
- يظهر الإطار المتواجد على يسار صندوق القائمة المنسدلة ليحتوي على بيانات هذا المركب.
- يحدث انكماش لعنصر DataGrid لتظهر بيانات هذا العميل فقط.

كما يستطيع المستخدم أن يضغط على الزر "List all customers" لكي تظهر بيانات جميع العملاء داخل شبكة البيانات، ويستطيع أيضاً الضغط على الزر "Add Customer And Boat" لكي يظهر صفحة ويب أخرى تمكّن المستخدم من إضافة كل من بيانات عميل جديد ومركب جديد. يستخدم العنصر lblNbrUsers لإظهار عدد مرات زيارة الصفحة. يوضح الشكل رقم (١٦,٢٧) الصفحة الرئيسة قبل اختيار عميل محدد.

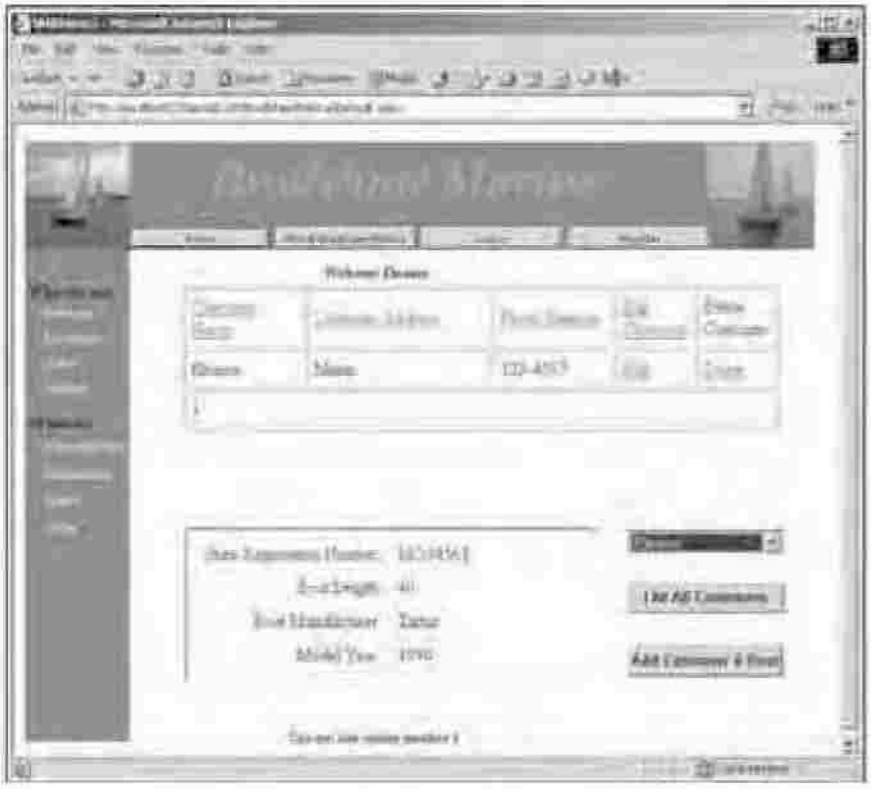
يشمل الجزء العلوي من صفحة الويب على عنصر User Control الذي يحتوي على أربعة أزرار أفقية للتجول (سوف نطور الثلاثة الأولى فقط خلال هذا الفصل). يوضح الشكل رقم (١٦,٢٨) الصفحة الرئيسة بعد اختيار العميل Eleanor من القائمة المنسدلة حيث تظهر بيانات المركب الخاص به في إطار بيانات المركب وتقلصت شبكة البيانات لتظهر بياناته فقط.

والآن نعود للوضع الأول وذلك بالضغط على الزر "List all customers" لعرض بيانات جميع العملاء وإخفاء إطار بيانات المركب، وعندئذ يمكننا تعديل بيانات أحد العملاء وذلك بالضغط على الرابط Edit المقابل للعميل الذي نريد أن نعدل بياناته فيحدث ما يلي:

- تظهر صناديق حوارية نصية تحتوي على كل من الاسم والعنوان.
- يحل محل الرابط Edit كل من الرابط Update والرابط Cancel.

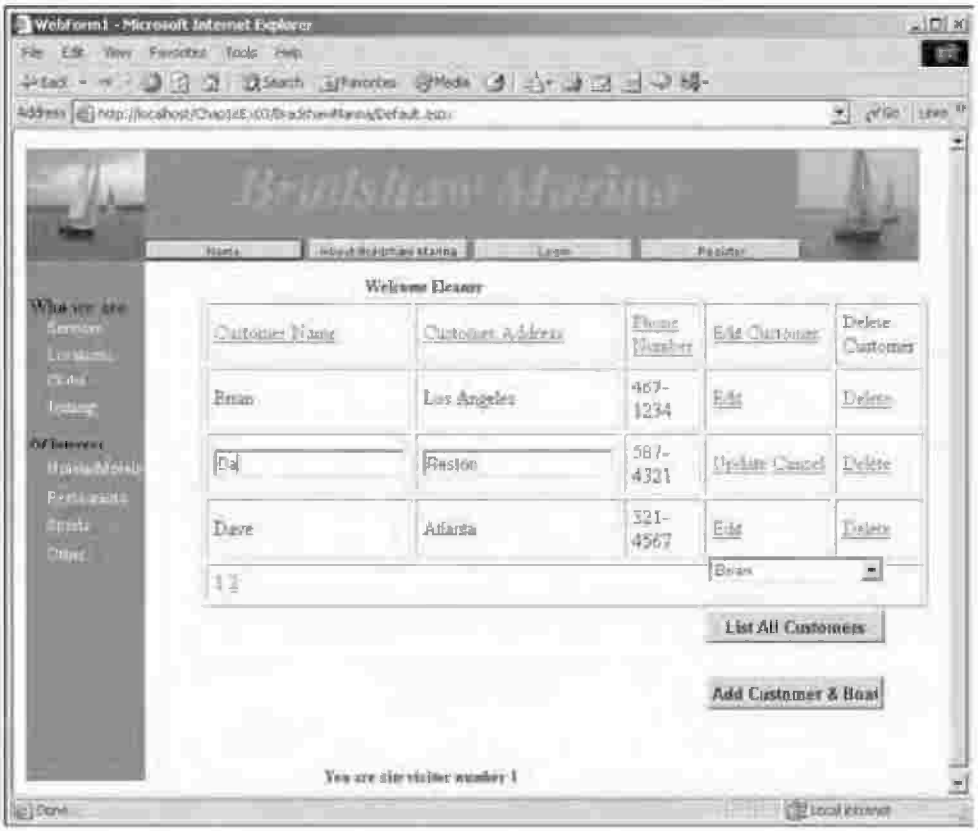


الشكل رقم (١٦,٢٧). الصفحة الرئيسة قبل اختيار عميل محدد.



الشكل رقم (١٦,٢٨). الصفحة الرئيسة بعد اختيار العميل.

يوضح الشكل رقم (١٦,٢٩) هذه النتيجة بعد الضغط على الرابط Edit، ولاحظ أن رقم الهاتف يمثل المفتاح الأساسي؛ ولذلك لا يمكن تعديله، ومن ثم لا يظهر صندوق نص أمام العمود "Phone Number" (قد تم إنجاز ذلك بالتأشير على الاختيار "Read Only" لهذا العمود من الصفحة Columns داخل نافذة خصائص عنصر شبكة البيانات).



الشكل رقم (١٦,٢٩). تعديل العميل في Data Grid.

والآن جرب أن تعدل اسم العميل "Dan" وذلك بحذف الحرف الأخير ليصبح "Da"، ثم اضغط على الرابط Update فعندئذ سيتغير اسم العميل "Da" داخل قاعدة البيانات (يمكن التحقق من ذلك بفحص جدول Customers المتواجد داخل قاعدة البيانات)، كما يمكنك الضغط على الرابط Delete لحذف بيانات عميل ما.

شرح شفرة تطوير صفحة الويب الرئيسة

Examining the Code for the Default Web Form

وللتعامل مع قاعدة البيانات والتعامل مع المصفوفات نحتاج إلى تنفيذ أمر Import للفضاء المسمى الخاص بكل منها، وكذلك نحتاج إلى تعريف بعض متغيرات الإشارة للاتصال بقاعدة البيانات والمصفوفات، وتقوم الأوامر

التالية بتنفيذ تلك المهام، ولدعم مفهوم كل من فهرسة البيانات وتصنيفها داخل شبكة البيانات نحتاج إلى تعريف بعض المتغيرات داخل الأوامر التالية:

```
Imports System.Data.OleDb
Imports System.Collections
Public Class WebForm1
    Inherits System.Web.UI.Page

    ' Declare a connection
    Dim aConnection As New OleDbConnection()

    ' Declare a Dataview
    Dim dv As New DataView()

    ' Declare variables for customer and boat instances
    Dim aCustomer As Customer
    Dim aBoat As Boat

    ' Declare an ArrayList for the customer instances
    Shared customers As ArrayList
    ' Declare variables for the sort key and page index
    Shared SK As String = "Name"
    Shared NP As Integer = 0
```

والآن يمكننا إنجاز الاتصال بقاعدة البيانات وذلك بكتابة الأوامر التالية داخل إجراء معالجة الحدث Page_Init حيث يجب عليك أولاً إيجاد هذا الإجراء وذلك بتوسعة منطقة الأوامر المنتجة تلقائياً بواسطة مصمم النماذج داخل نافذة محرر الأوامر.

```
Private Sub Page_Init(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles MyBase.Init
        Handles MyBase.Init
        'CODEGEN: This method call is required by the Web Form Designer
        'Do not modify it using the code editor.
        InitializeComponent()

        aConnection = CustomerAndBoatDatabaseConnect.Initialize()
        Customer.Initialize(aConnection)
        Boat.Initialize(aConnection)
       End Sub
```

ثم يأتي إجراء معالجة الحدث On-load الذي يمكن أن يحتوي على الأوامر التي نريد تنفيذها عند أول مرة تحميل للصفحة وليس عند الاستدعاءات التالية. يوضح الشكل رقم (١٦,٣٠) الإجراء Page_Load المسؤول عن

استرجاع بيانات العملاء من قاعدة البيانات وإسنادها للمصفوفة Customers، ثم استدعاء الإجراء BindToGrid الذي ينشئ جدولاً من بيانات العملاء باستخدام المصفوفة Customers، ثم ربطه بشبكة البيانات.

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

    If Not IsPostBack Then

        customers = Customer.GetAll
        ' Invoke the bind method
        BindToGrid(customers, Nothing, 0)

        ' Add to number of site visits
        Application.Lock()
        Application("nbrUsers") += 1
        Application.UnLock()
        lblNbrUsers.Text = "You are site visitor number " & _
            Application("nbrUsers")
        ' Get customer cookie if it exists
        Try
            Dim cName As String = Request.Cookies("Name").Value
            Dim cAddress As String = Request.Cookies("Address").Value
            lblReturningCustomer.Text = "Welcome " & cName
        Catch ce As Exception
            lblReturningCustomer.Text = " "
        End Try
    End If
    ' Make panel and controls invisible-----------------------

    pnlBoat.Visible = False

    lblBtReg.Visible = False
    lblBtLength.Visible = False
    lblBtYear.Visible = False
    lblBtManufacturer.Visible = False
    lblStateReg.Visible = False
    lblBoatLength.Visible = False
    lblManufacturer.Visible = False
    lblYear.Visible = False

    ' ------------------------------------------------------------

End Sub
```

الشكل رقم (١٦,٣٠). الإجراء Page_Load.

ثم يتم زيادة متغير التطبيق nbrUsers بمقدار واحد الذي يعبر عن عدد مرات زيارة الصفحة الرئيسة، وتذكر أهمية استدعاء كل من الإجراء Lock والإجراء Unlock قبل زيادة هذا المتغير وبعده وذلك لأن هذا المتغير عام على مستوى جميع زوار الموقع.

ثم يقوم الإجراء باسترجاع محتويات ملف Cookie إذا كان موجوداً وإسناده إلى العنصر lblReturningCustomer الذي يظهر أعلى شبكة البيانات. أما الجزء الأخير من الإجراء فيهتم بإخفاء كل من إطار بيانات المركب والعناوين المسؤولة عن إظهار تلك البيانات. أما الإجراء BindToGrid المسؤول عن ربط بيانات العملاء بشبكة البيانات فهو موضح في الشكل رقم (١٦,٣١). يتشابه هذا الإجراء مع إجراء الربط السابق ولكن مع الفرق أن المستخدم هنا هو بيانات العملاء بدلاً من بيانات المراكب، كما أن البيانات يتم استرجاعها من قاعدة بيانات.

وعندما يتم اختيار اسم أحد العملاء من القائمة المنسدلة يتم استخدام هذا الاسم لاسترجاع بيانات المركب الخاص بذلك العميل كما هو واضح في الشكل رقم (١٦,٣٢) الذي يشمل على إجراء معالجة حدث اختيار القائمة المنسدلة.

لاحظ أهمية إسناد القيمة True للصفة AutoPostBack للقائمة المنسدلة وذلك لضرورة معالجة أحداث عناصر الخادم على الخادم مباشرة.

يهتم الجزء الأول من هذا الإجراء بإظهار كل من إطار بيانات المركب والصناديق النصية الخاصة بعرض هذه البيانات.

ولأنه ربما يوجد اختلاف بين كل من الفهرس الخاص بالقائمة المنسدلة وفهرس المصفوفة، فمن المفضل البحث عن العميل برقم الهاتف المسترجع من القائمة المنسدلة للعميل الحالي باستخدام التكرار For-Next حيث يتم مقارنة رقم هاتف عملاء المصفوفة برقم هاتف العميل الحالي إلى أن نجد هذا العميل ومن ثم يمكن استعادة بيانات المركب للعميل الحالي وإظهارها على عناصر العناوين الخاصة بهم.

أما الجزء الأخير من هذا الإجراء فهو مسؤول عن مسح محتويات المصفوفة Customers وإضافة العميل الحالي لنفس المصفوفة، ثم ربطها بشبكة البيانات باستخدام إجراء الرابط BindToGrid نفسه لتكون النتيجة هي إظهار بيانات هذا العميل فقط داخل شبكة البيانات.

أما الأوامر التالية لإجراء معالجة الزر "List All Customers" فهي مسؤولة أولاً عن إخفاء إطار بيانات المركب والعناوين الخاصة بذلك، ثم استدعاء الإجراء GetAll لاسترجاع جميع العملاء وإسنادها للمصفوفة Customers التي يتم ربطها بشبكة البيانات باستخدام الإجراء BindToGrid.

```
Private Sub btnAllCust_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnAllCust.Click

    ' Make panel and controls visible-------------------------------------

    pnlBoat.Visible = False
    lblBtReg.Visible = False
    lblBtLength.Visible = False
    lblBtYear.Visible = False
    lblBtManufacturer.Visible = False
    lblStateReg.Visible = False
    lblStateReg.Visible = False
    lblBoatLength.Visible = False
    lblManufacturer.Visible = False
    lblYear.Visible = False

    ' --------------------------------------------------------------------

    customers = Customer.GetAll
    ' Invoke the bind method
    BindToGrid(customers, SK, NP)

End Sub
```

```
Private Sub BindToGrid(ByVal customers As ArrayList, _
    Optional ByVal sortkey As String = "name", _
    Optional ByVal newpage As Integer = 0)

    Dim myCust As Customer
    Dim dt As New DataTable()
    Dim dr As DataRow
    dt.Columns.Add(New DataColumn("Name", GetType(String)))
    dt.Columns.Add(New DataColumn("Address", GetType(String)))
    dt.Columns.Add(New DataColumn("PhoneNo", GetType(String)))

    For Each myCust In customers
        dr = dt.NewRow

        dr("Name") = myCust.GetName
        dr("Address") = myCust.GetAddress
        dr("PhoneNo") = myCust.GetPhoneNo

        dt.Rows.Add(dr)
    Next

    dv = dt.DefaultView

    If sortkey = Nothing Then sortkey = "name"
    If newpage = Nothing Or newpage = 0 _
        Or newpage = DataGrid1.PageCount + 1 Then
        newpage = 0
    End If

    dv.Sort = sortkey
    DataGrid1.CurrentPageIndex = newpage

    DataGrid1.DataSource = dv
    DataGrid1.DataBind()

    cblName.DataSource = dv
    cblName.DataTextField = "Name"
    cblName.DataValueField = "PhoneNo"
    cblName.DataBind()

End Sub
```

الشكل رقم (١٦,٣١). الإجراء BindToGrid.

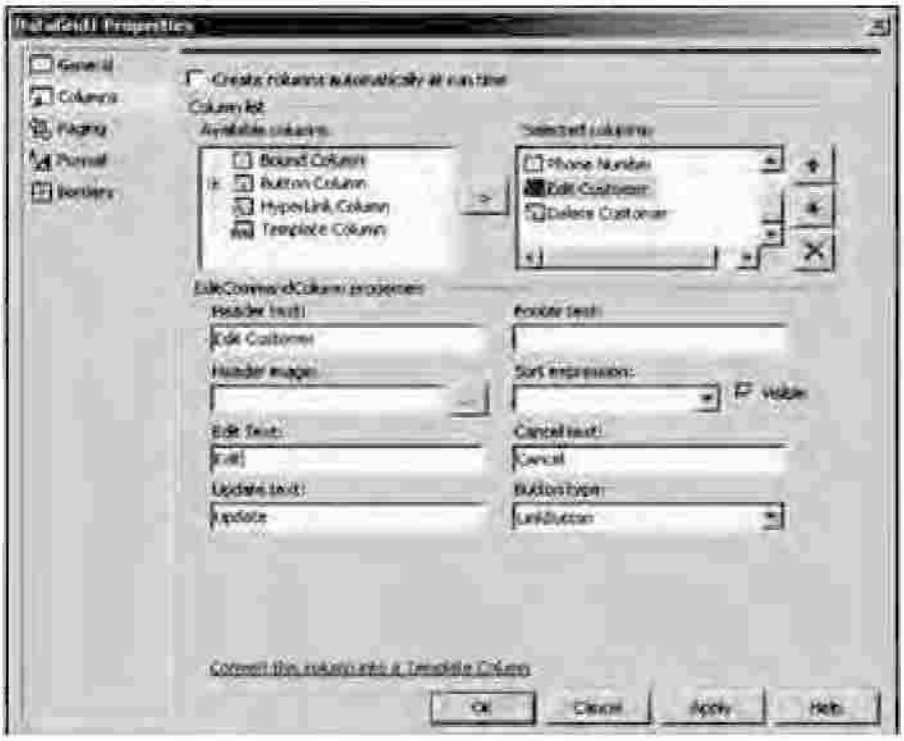
```
Private Sub ddlNames_SelectedIndexChanged(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles ddlNames.SelectedIndexChanged
        Dim i As Integer
        ' Make panel and controls visible-----------------------------

        pnlBoat.Visible = True
        lblStReg.Visible = True
        lblStLength.Visible = True
        lblStYear.Visible = True
        lblStManufacturer.Visible = True
        lblStateReg.Visible = True
        lblBoatLength.Visible = True
        lblManufacturer.Visible = True
        lblYear.Visible = True

        ' ------------------------------------------------------------

        i = 0
        Dim nPhoneNo As String = ddlNames.SelectedItem.Value
        Dim cust As Customer
        For Each cust In customers
            If cust.GetPhoneNo = nPhoneNo Then
                ' Do nothing but get index value of ArrayList
                Exit For
            End If
            i = i + 1
        Next

        aCustomer = customers(i)

        aBoat = aCustomer.GetBoat

        lblStateReg.Text = aBoat.GetStateRegistrationNo
        lblBoatLength.Text = aBoat.GetLength
        lblManufacturer.Text = aBoat.GetManufacturer
        lblYear.Text = aBoat.GetYear

        customers.Clear()
        customers.Add(aCustomer)
        BindToGrid(customers, SK, 0)

    End Sub
```

الشكل رقم (١٦,٣٢). إجراء معالجة حدث اختيار القائمة المنسدلة.

أما أوامر إجراء معالجة الزر "Add Customer & Boat" فهي مسؤولة عن إظهار نموذج إضافة بيانات عميل ومركب؛ ولذلك سيتم استخدام الإجراء Redirect المعرف داخل الصنف Response لاستدعاء الملف AddCustomerBoat.aspx كما هو واضح من الأوامر التالية:

```
Private Sub btnAddCustBoat_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles btnAddCustBoat.Click

        Response.Redirect("AddCustomerBoat.aspx")
End Sub
```

وكما رأينا أن شبكة البيانات تحتوي على العمود Edit والعمود Delete اللذين يمكن إضافتهما باستخدام نافذة "Property Builder"، وعلى أي حال فإن كلاً من هذين العمودين ينتميان للنوع "Bound Columns" (انظر الشكل رقم ١٦،٣٣).



الشكل رقم (١٦،٣٣). إلغاء تحديد عمود Data Grid.

ولتعريف إجراء التعديل والحذف والإلغاء، يجب أولاً الإشارة لإجراءات معالجة الأحداث Edit و Update داخل أوامر شفرة HTML الخاصة بشبكة البيانات (<asp:datagrid>) كما هو واضح من الأوامر التالية :

```
<asp:datagrid
        OnUpdateCommand="DataGrid1_Update"
        OnDeleteCommand="DataGrid1_Delete"
        OnCancelCommand="DataGrid1_Cancel"
        OnEditCommand="DataGrid1_Edit"
        DataKeyField="phoneNo"
```

لاحظ أن أسماء الإجراءات تم إستدعاها بواسطة المبرمج وهي الأسماء التي تبدأ بالعبارة "DataGrid1"، أما الاسم PhoneNo فيمثل اسم حقل بيانات. توضح الأوامر التالية إجراء معالجة الحدث Edit حيث يتم إرسال صف

شبكة البيانات (الصف محل التعديل) إلى هذا الإجراء الذي يستخلص الفهرس الخاص به، ثم يسنده للخاصية EditItemIndex لشبكة البيانات، ثم يستدعي الإجراء BindToGrid لإجراء عملية الربط. وكما ذكرنا سابقاً أن الرابط Edit يتغير إلى كل من الرابط Update والرابط Cancel تلقائياً بعد الضغط على الرابط Edit، ومن ثم يستطيع المستخدم تعديل بيانات الصنف الحالي ثم الضغط على الرابط Update.

```
Sub DataGrid1_Edit(ByVal sender As Object, _
     ByVal e As DataGridCommandEventArgs)

     DataGrid1.EditItemIndex = e.Item.ItemIndex
     ' Reference row, do update, invoke bind
     BindToGrid(customers, SK, NP)
    End Sub
```

والآن يمكننا تفحص أوامر إجراء معالجة الحدث Update المسؤول عن تعديل بيانات صف شبكة البيانات الحالي داخل قاعدة البيانات (توضح الأوامر التالية هذا الإجراء). تذكر أن البيانات تتحول إلى صناديق نصية لتمكّن المستخدم من تعديلها (عدا رقم هاتف العميل الذي يمثل مفتاحاً أساسياً للعملاء). ويقوم هذا الإجراء بتعريف متغيرات إشارة للصناديق النصية (Text Boxes) واسترجاع القيم المدخلة بواسطة المستخدم عدا رقم الهاتف الذي لم يتحول إلى صندوق نص، ومن ثم يتم استرجاعه من خلية الجدول (Table Cell) مباشرة.

```
Sub DataGrid1_Update(ByVal sender As Object, _
ByVal e As DataGridCommandEventArgs)

Dim nameText As TextBox = CType(e.Item.Cells(0).Controls(0), TextBox)
Dim addressText As TextBox = CType(e.Item.Cells(1).Controls(0), TextBox)
Dim phoneCell As TableCell = e.Item.Cells(2)

Dim name As String = nameText.Text
Dim address As String = addressText.Text
Dim phoneNo As String = phoneCell.Text

aCustomer = New Customer(name, address, phoneNo)
aCustomer.update()
customers = Customer.GetAll()

DataGrid1.EditItemIndex = -1
BindToGrid(customers, SK, NP)

End Sub
```

ثم يقوم الجدول بإنشاء كائن من الصنف Customer واستدعاء الإجراء Update لتعديل بياناته في قاعدة البيانات، ثم يقوم الإجراء بإسناد القيمة "-١" للخاصية EditItemIndex لإلغاء وضع التعديل واستبدال كل من الرابط Update والرابط Cancel بالرابط Edit مرة ثانية، ثم يقوم الإجراء باستدعاء الإجراء BindToGrid لإجراء الربط بين بيانات العملاء بقاعدة البيانات وشبكة البيانات. يتطلب إجراء معالجة الحدث Cancel إلغاء وضع التعديل فقط كما هو واضح من الأوامر التالية:

```
Sub DataGrid1_Cancel(ByVal sender As Object, _
     ByVal e As DataGridCommandEventArgs)
     DataGrid1.EditItemIndex = -1
     BindToGrid(customers, SK, NP)
 End Sub
```

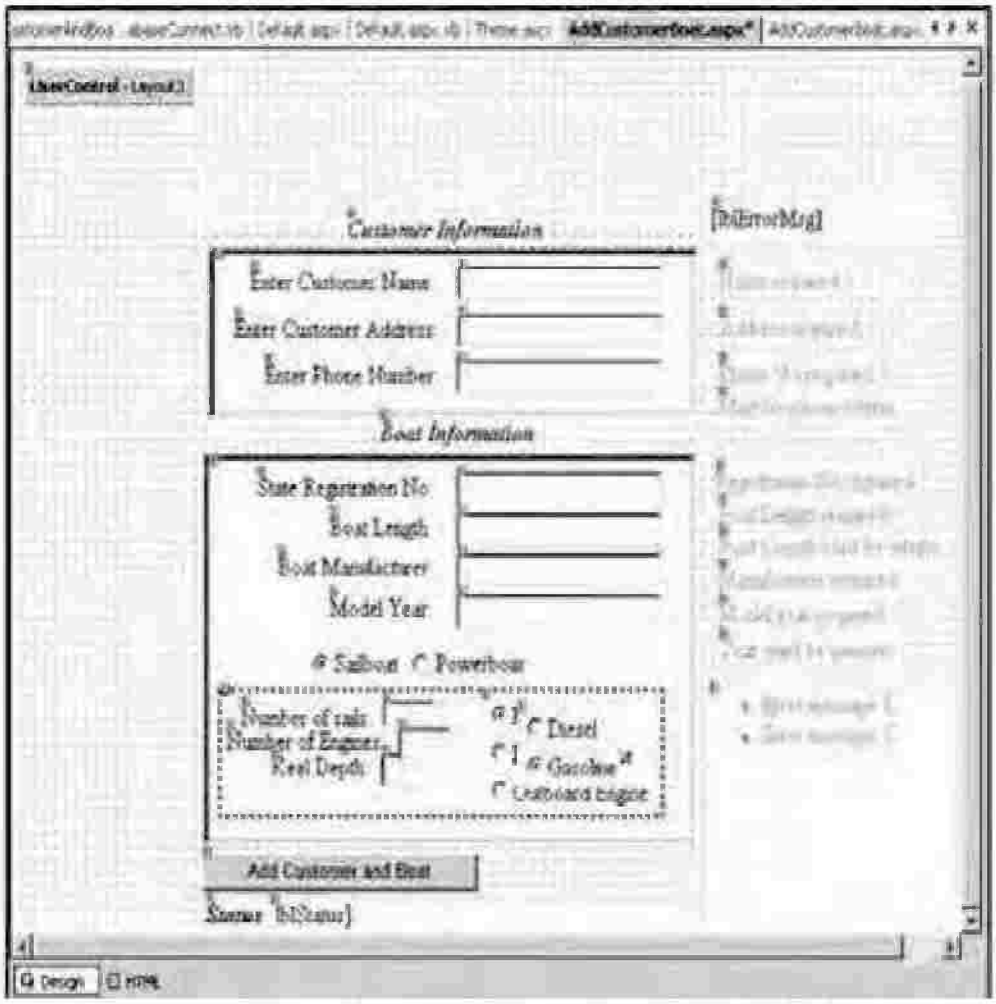
أما إجراء معالجة الحدث Delete فهو يشبه إجراء معالجة الحدث Update عدا أن البيانات يتم استرجاعها من خلايا الصف مباشرة بدلاً من استرجاعها من صناديق نصية واستدعاء الإجراء Delete من كائن الصنف Customer بدلاً من الإجراء Update كما هو واضح من الأوامر التالية:

```
Sub DataGrid1_Delete(ByVal sender As Object, _
        ByVal e As DataGridCommandEventArgs)

        Dim nameCell As TableCell = e.Item.Cells(0)
        Dim addressCell As TableCell = e.Item.Cells(1)
        Dim phoneCell As TableCell = e.Item.Cells(2)

        Dim name As String = nameCell.Text
        Dim address As String = addressCell.Text
        Dim phoneNo As String = phoneCell.Text

        aCustomer = New Customer(name, address, phoneNo)
        aCustomer.Delete()
        customers = Customer.GetAll()
        BindToGrid(customers, SK, NP)
        End Sub
```

تصميم نموذج الويب AddCustomerBoat
Designing the AddCustomerBoat Web Form

يتطلب نموذج ويب إضافة بيانات عميل جديد ومركب جديد صناديق نصية لاستقبال البيانات من المستخدم كما هو واضح من الشكل رقم (١٦,٣٤)، ولأن النموذج مسؤول عن استقبال بيانات فيجب التحقق من صحتها باستخدام عناصر التحقق من صحة البيانات حيث تظهر رسائل الخطأ على يمين الصناديق النصية عند

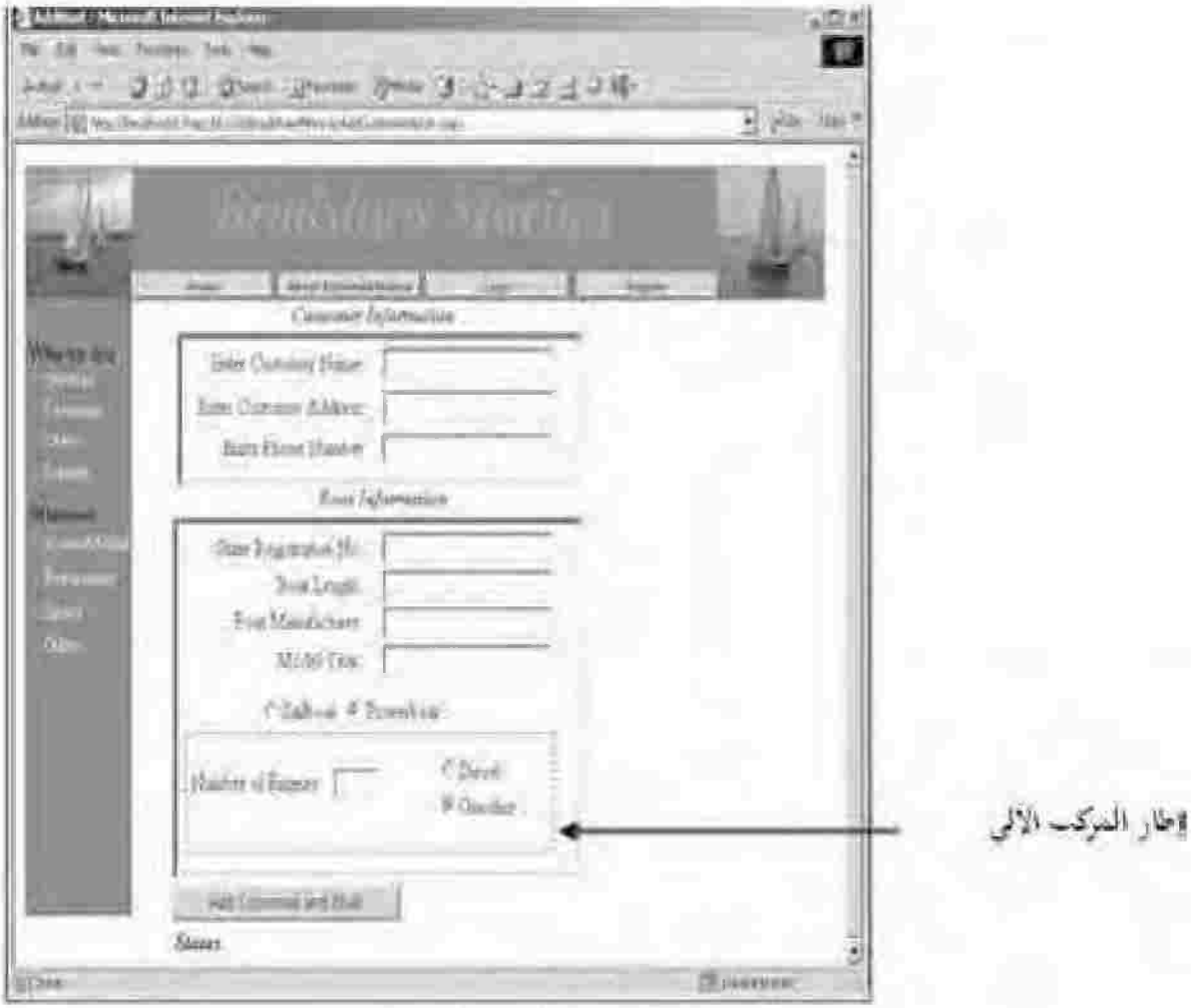
إدخال قيم غير صحيحة بها (راجع الفصل الثاني عشر)، كما يوجد ملخص لرسائل الخطأ الخالية أسفل عمود الرسائل المتعددة حيث يستخدم كل عنصر من النوع Text Box عنصر تحقق من صحة البيانات من النوع RequiredFieldValidation.



الشكل رقم (١٦,٣٤). نموذج ويب إضافة بيانات عميل جديد ومركب جديد.

توجد هناك ثلاثة عناصر تحتاج إلى تحقق أكثر من صحة البيانات وهي: طول المركب (Boat Length)، وسنة الموديل (Model Year)، ورقم التليفون (Phone Number) حيث يجب أن يكون طول المركب من النوع Double، ويجب أن يكون نوع السنة Integer، ويجب أن يحتوي رقم الهاتف على أرقام فقط ويتبع الشكل المعروف لأرقام الهاتف (ثلاثة أرقام، ثم الرمز "-"، ثم ثلاثة أرقام، ثم الرمز "-"، ثم أربعة أرقام)؛ ولذلك يمكن استخدام عنصر التحقق من صحة البيانات CompareValidator للتحقق من صحة البيانات المدخلة، ومن ثم سيصاحب كل

عنصر إدخال من هذه العناصر الثلاثة عنصرين للتحقق من صحة البيانات (العنصر RequiredFieldValidator والعنصر CompareValidator)، أما الجزء السفلي من النموذج الذي يشمل على بيانات المركب فيحتوي على إطارين (Two Panels) حيث يحتوي الإطار الأول على البيانات الخاصة بالمركب الشراعي (Sailboat) والآخر يحتوي على البيانات الخاصة بالمركب الآلي (Powerboat)، ولاحظ أن النموذج يحتوي على قائمة اختيار نوع المركب باستخدام أزرار الاختيار (Radio Buttons)، حيث يظهر إطار بيانات المركب الشراعي عند ظهور النموذج في البداية (انظر الشكل رقم ١٦،٣٥)، كما يستطيع المستخدم أن يغير نوع المركب إلى المركب الآلي بالضغط على الاختيار Powerboat وعندئذ سيظهر إطار بيانات المركب الآلي (انظر الشكل رقم ١٦،٣٦).



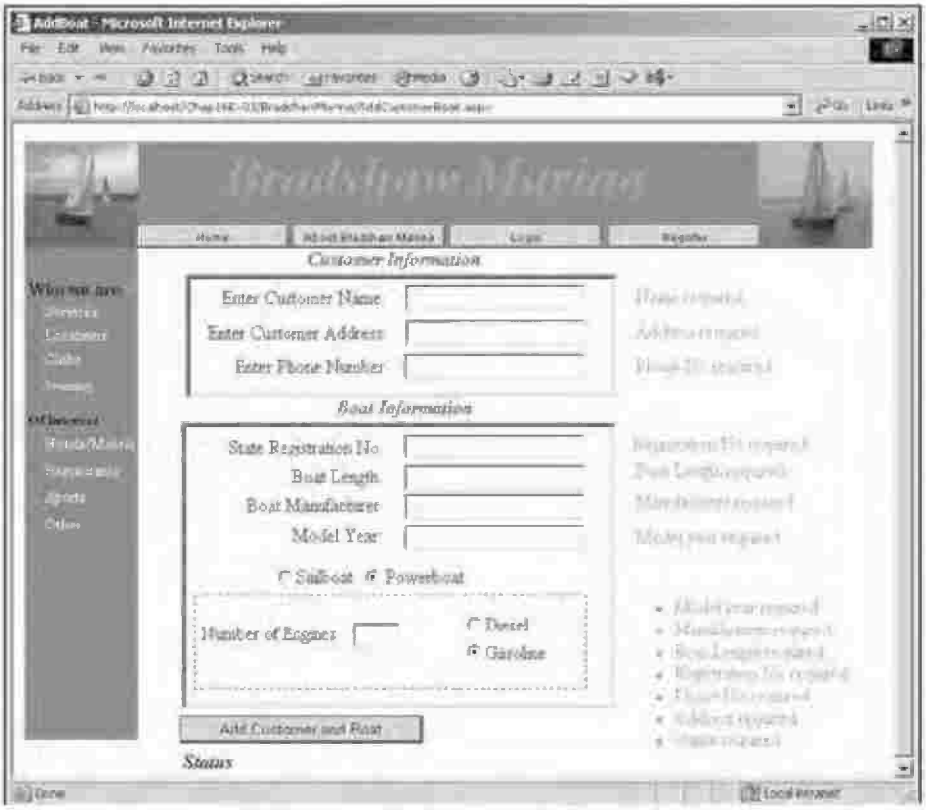
الشكل رقم (١٦،٣٥). تغيير نوع المركب إلى المركب الشراعي

الشكل رقم (١٦،٣٦). تغيير نوع المركب إلى المركب الآلي.

يوضح الشكل رقم (١٦،٣٧) كيفية عمل عناصر التحقق من صحة البيانات حيث يتم استدعاء إجراء التحقق من صحة البيانات بواسطة تلك العناصر عند الضغط على أحد أزرار نموذج الويب. كما يوضح الشكل رقم (١٦،٣٧) الضغط على زر "Add Customer And Boat" وعدم إدخال أي بيانات ومن ثم ظهرت جميع رسائل الخطأ على يمين الحقول، ويظل النموذج هكذا إلى أن يدخل المستخدم بيانات صحيحة، ثم يضغط على زر الإضافة أو يضغط على زر Back المتواجد أعلى برنامج متصفح الويب.

تظهر جميع رسائل الخطأ المنفردة وملخص رسائل الخطأ على يمين حقول النموذج، كما تظهر رسائل خطأ التحقق الثاني للحقول التي تتطلب تحققاً مزدوجاً مثل حقل رقم الهاتف على اليمين أيضاً (حاول أن تقوم بإدخال حروف بدل الأرقام في حقل رقم الهاتف).

والآن تم بإضافة بيانات عملاء جدد ومراكب جدد خاصة بهم، ثم اضغط على الزر Home للعودة للصفحة الرئيسة، وعندئذ يظهر العملاء الجدد داخل شبكة البيانات، وكلما تم إضافة عملاء جدد زاد عدد صفحات شبكة البيانات.



الشكل رقم (١٦,٣٧). عناصر التحقق من صحة البيانات.

أوامر نموذج الويب AddCustomerBoat

Code for the AddCustomerBoat Web Form

يجب تعريف متغير إشارة من النوع Customer وتعريف ثلاثة متغيرات من النوع String كما هو واضح من الأوامر التالية:

```
Public Class AddBoat
    Inherits System.Web.UI.Page

    ' Declare customer reference variable
    Private aCustomer As Customer
    ' Declare string variables for name, address and phone
    Private customerName, customerAddress, customerPhone As String
```

توضح الأوامر التالية إضافة بيانات كل من عميل جديد ومركب جديد حيث يجب تعريف مجموعة من المتغيرات أولاً للاحتفاظ ببيانات كل من العميل والمركب، ثم التحقق من أن الصفة IsValid للصنف Page تساوي True مما يعني أن البيانات المدخلة صحيحة، ومن ثم تقوم الأوامر بتعريف كائن من الصنف Customer واستدعاء

الإجراء AddNew لإضافته في قاعدة البيانات، وبالمثل يمكن إضافة بيانات المركب ولكن يجب في البداية معرفة هل المركب المختار حالياً هو مركب شراعي أم مركب آلي.

```
Private Sub btnAdd_Click(ByVal sender As System.Object, _
        ByVal e As System.EventArgs) Handles btnAdd.Click

  Dim customerName, customerAddress, CustomerPhone As String
  Dim stateRegNo, manufacturer As String
  Dim boatLength, modelYear As Integer



  If Page.IsValid Then
      aCustomer = New Customer(customerName, customerAddress, CustomerPhone)
      lblStatus.Text = aCustomer.GetName
      If rblBoatType.Items(0).Selected Then
          addSailboat(stateRegNo, boatLength, manufacturer, modelYear)
      ElseIf rblBoatType.Items(1).Selected Then
          addPowerboat(stateRegNo, boatLength, manufacturer, modelYear)
      End If
  End If
End Sub
```

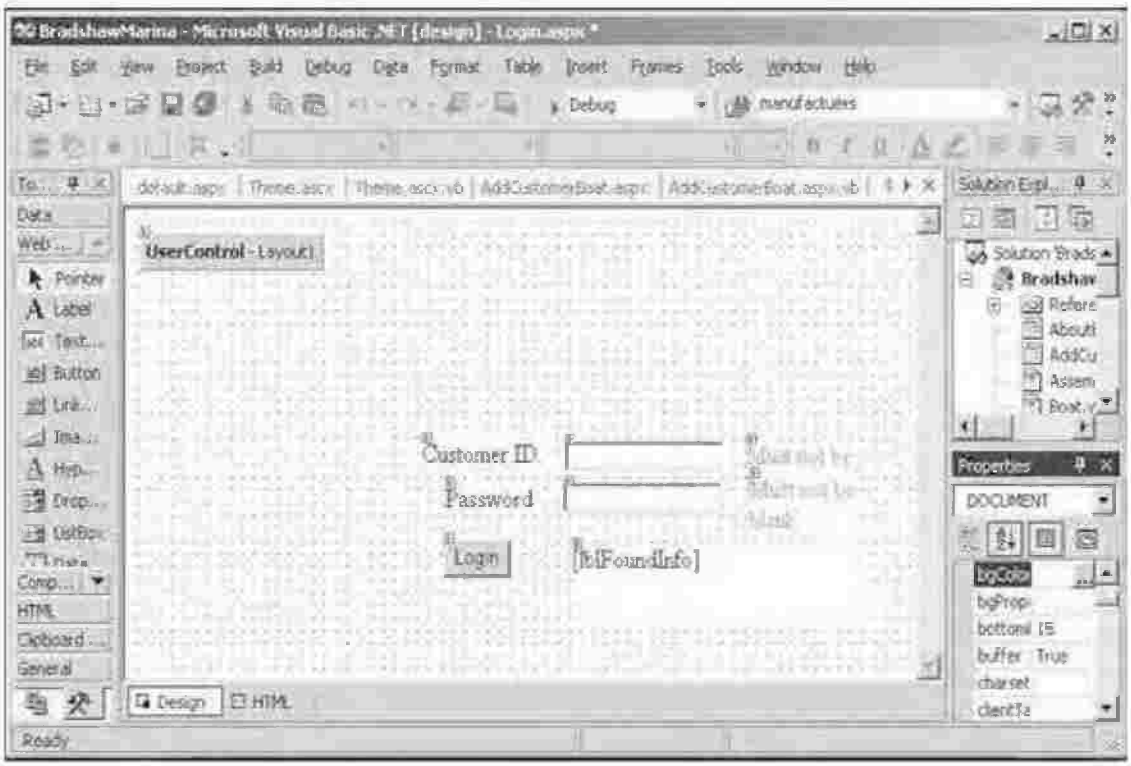
تذكر أن قائمة أزرار الاختيار تستخدم لتفعيل إطار المركب الشراعي أو تفعيل إطار المركب الآلي؛ ولذلك يوضح إجراء معالجة أزرار الاختيار التالي كيفية عمل ذلك:

```
Private Sub rblBoatType_SelectedIndexChanged(ByVal sender As System.Object, _
ByVal e As System.EventArgs) Handles rblBoatType.SelectedIndexChanged

        If rblBoatType.Items(0).Selected Then
            ' Sail Boat
            pnlPowerboat.Visible = False
            lblNumberOfEngines.Visible = False
            txtNumberOfEngines.Visible = False
            rblFuel.Visible = False

            pnlSailBoat.Visible = True
            lblNumberOfSails.Visible = True
            lblKeelDepth.Visible = True
            txtNumberOfSails.Visible = True
            txtKeelDepth.Visible = True
            rblEngine.Visible = True
        Else
            ' Powerboat
            pnlPowerboat.Visible = True
            lblNumberOfEngines.Visible = True
            txtNumberOfEngines.Visible = True
            rblFuel.Visible = True
```

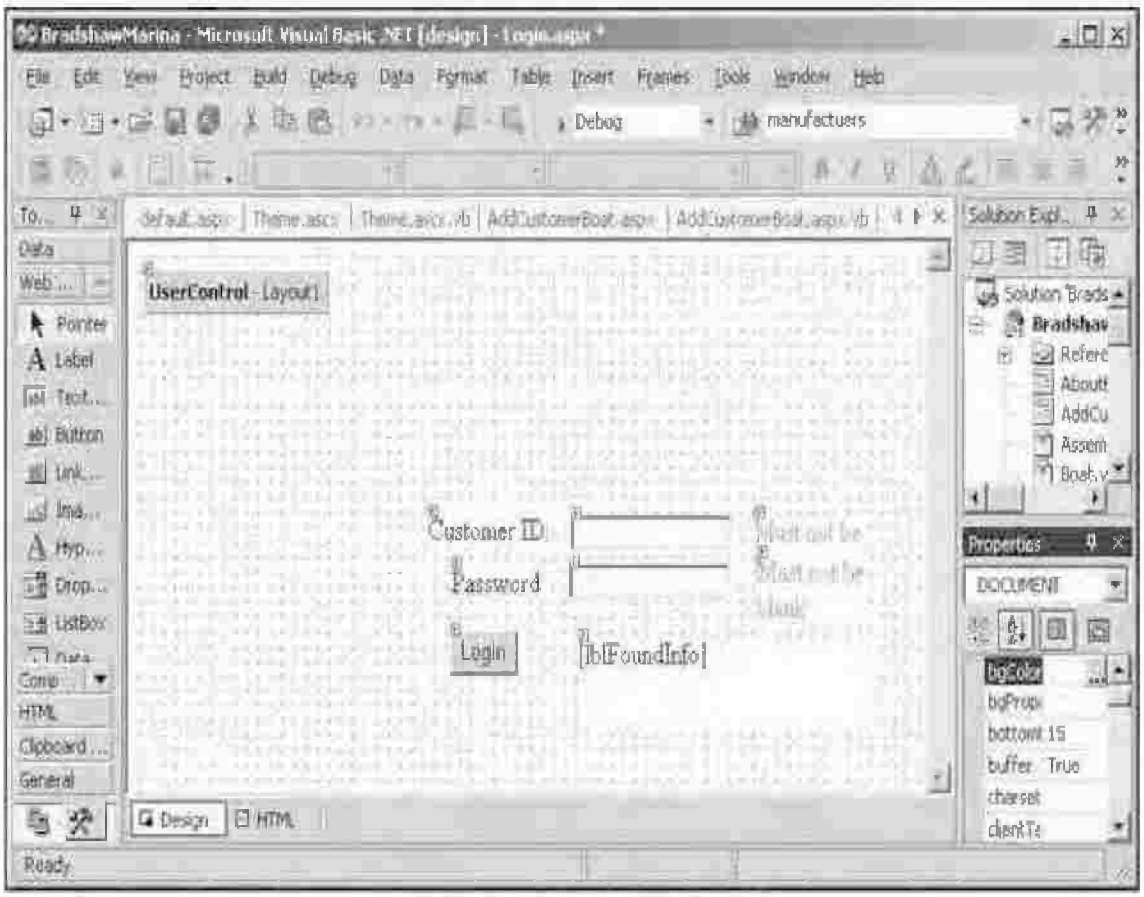
```
            pnlSailBoat.Visible = False
            lblNumberOfSails.Visible = False
            lblKeelDepth.Visible = False
            txtNumberOfSails.Visible = False
            txtKeelDepth.Visible = False
            rblEngine.Visible = False

        End If
    End Sub
```

تصميم نموذج الدخول للموقع
Designing the Startup Web Form

يوضح الشكل رقم (١٦،٣٨) نموذج الدخول للموقع (login Web form) الذي يحتوي على رقم العميل (Customer ID) الممثل في رقم هاتف العميل وكلمة السر، كما يحتوي النموذج على عناصر التحقق من صحة البيانات. عند زيارة العميل الموقع يجب أن يزور أولاً نموذج الدخول ليدخل رقم الهاتف وكلمة السر، فإذا كانت البيانات صحيحة ستظهر رسالة ترحيب بالمستخدم كما هو واضح في الشكل رقم (١٦،٣٩)، كما سيتم إنشاء ملف Cookie ليحتفظ باسم العميل وكلمة السر الخاصة به والذي ستنتهي مدة صلاحيته بعد ثلاثة شهور من الآن.



الشكل رقم (١٦،٣٨). نموذج الدخول للموقع.



الشكل رقم (١٦,٣٩). نموذج الدخول الناجح.

عادة ما يتم تخزين كل من اسم المستخدم وكلمة السر في جدول منفصل، ومن ثم يتم فحص كل من الجدولين، ولكن في هذا المثال البسيط سيتم التحقق من رقم الهاتف فقط.

تصميم نموذج حول شركة برادشو ماريتا

Designing the AboutBradshawMarina Web Form

عندما يضغط المستخدم على "About Bradshaw Marina"، سيتم فتح صفحة ويب ثابتة المحتوى والموضحة في الشكل رقم (١٦,٤٠). أما زر Register فيؤدي إلى نموذج ويب مسؤول عن تسجيل زائر جديد للموقع (لم يتم تطوير هذا الجزء).

عزيزي القارئ لاحظ أن هذا المثال يركز على تصميم نماذج الويب فقط، ولكن يجب أن تنتبه ما كان لهذا الموقع أن يعمل إلا في وجود كل من أصناف مجال المشكلة وأصناف التعامل مع البيانات.

الشكل رقم (١٦,٤٠). صفحة ويب ثابتة المحتوى.

مقدمة عن لغة XML وخدمات الويب

Introduction to XML and Web Services

إن لغة XML اختصار للعبارة Extensible Markup Language والتي أصبحت أكثر المواضيع مناقشة خلال السنوات الأخيرة وذلك لأهميتها وقدرتها على تقديم وسيلة فعالة للتكامل بين مصادر بيانات مختلفة سواء داخل مؤسسة واحدة أو بين عدة تطبيقات لمؤسسات مختلفة عبر الإنترنت. إن معظم البيانات ذات الطبيعة المنظمة Structured Data يتم تخزينها داخل قواعد بيانات، ولكن توجد بيانات أخرى ذات طبيعة غير منظمة يتم تخزينها في أشكال أخرى (ملفات نصية، أو بريد إلكتروني، أو ملفات جداول إلكترونية، إلخ) مما أدى إلى أن معظم الشركات الموردة لقواعد البيانات أصبحت تدعم لغة XML لتخزين البيانات؛ ولذلك نوهت شركة مايكروسوفت أن جميع تطبيقاتها سوف تخزن البيانات بلغة XML بحلول عام ٢٠٠٦م.

لقد قدمت لغة XML لكي تحل مشاكل لغة HTML ومن أهم هذه المشاكل:

- لا تستطيع لغة HTML أن تعالج بيانات ذات طبيعة شجرية (Hierarchical Structure).
- لا توجد علاقة بين علامات لغة HTML ومحتواها.

- لا يستطيع مطورو تطبيقات الويب أن يتعاملوا مع علامات غير علامات HTML المعرفة مسبقاً والتي تمثل مجموعة ثابتة.

إن البيانات ذات الطبيعة الشجرية غالباً ما تحتاج أن تطورها داخل موقع الويب ولكن نجد صعوبة مع لغة HTML لمعالجة هذه البيانات، فعلى سبيل المثال لا تستطيع لغة HTML معالجة بيانات المركب، وعلى العكس فإن لغة XML تستخدم علامات تشبه علامات لغة HTML ولكن تستطيع أن تعالج مثل تلك البيانات ذات الطبيعة الشجرية. يوضح الشكل رقم (١٦,٤١) جزءاً من بيانات جدول المراكب بلغة XML.

```
<?xml version="1.0" standalone="yes" ?>
- <NewDataSet>
-  <Boats>
    <StateRegistrationNo>MO223344</StateRegistrationNo>
    <BoatLength>24</BoatLength>
    <Manufacturer>Tracker</Manufacturer>
    <Year>1996</Year>
    <CustomerPhoneNo>467-1234</CustomerPhoneNo>
</Boats>
-  <Boats>
    <StateRegistrationNo>MO457812</StateRegistrationNo>
    <BoatLength>19</BoatLength>
    <Manufacturer>Ranger</Manufacturer>
    <Year>2001</Year>
    <CustomerPhoneNo>587-4321</CustomerPhoneNo>
</Boats>
    .
    .
    .
</NewDataSet>
```

الشكل رقم (١٦,٤١). جزء من بيانات جدول المراكب بلغة XML.

وكما نرى من الشكل رقم (١٦,٤١) يقدم السطر الأول معلومات للتطبيق الذي يستقبل هذا الملف أن هذه الوثيقة مكتوبة بلغة XML الجيل رقم ١٠٥، مع العلم أن هذا الملف قد تم إنشاؤه آلياً باستخدام الإجراء WriteXML المعرف داخل الصنف DataSet مع إضافة الصفة "Standalone=yes". يجب أن توجد محتويات وثيقة XML داخل علامة لها بداية ونهاية (مثل العلامة <NewDataSet> والعلامة </NewDataSet>) وذلك مثل وثائق لغة HTML التي يجب أن تبدأ وتنتهي <HTML>.

وبالنظر إلى محتويات الوثيقة، نجد أن بيانات المركب توجد بشكل شجري حيث تظهر علامة <Boats>، ثم يظهر بداخلها بيانات مركب كامل حيث يوجد كل صفة داخل علامة خاصة بها مثل (<Year>1996</Year>)، ومع

أن لغة XML تستخدم علامات مثل لغة HTML فإن هذه العلامات ليست ثابتة، وهذا يعني أن علامات XML تقبل الزيادة من قبل المطورين.

ومن ثم فإن تقنية XML تقدم أسلوباً فعالاً للتكامل بين التطبيقات المختلفة سواء كانت في مؤسسة واحدة أم داخل عدة مؤسسات من خلال شبكة الإنترنت حيث يمكن الإيقاف على علامات محددة والتي تتألف منها البيانات ذات الطبيعة الشجرية بين التطبيقات المختلفة. كما تقدم تقنية XML العديد من الفوائد والأدوات الأخرى مثل لغة تعريف نوع الوثيقة (DTD)، ولغة تعريف تركيب البيانات (XSD) التي يمكن الاستفادة منها في هذا الغرض.

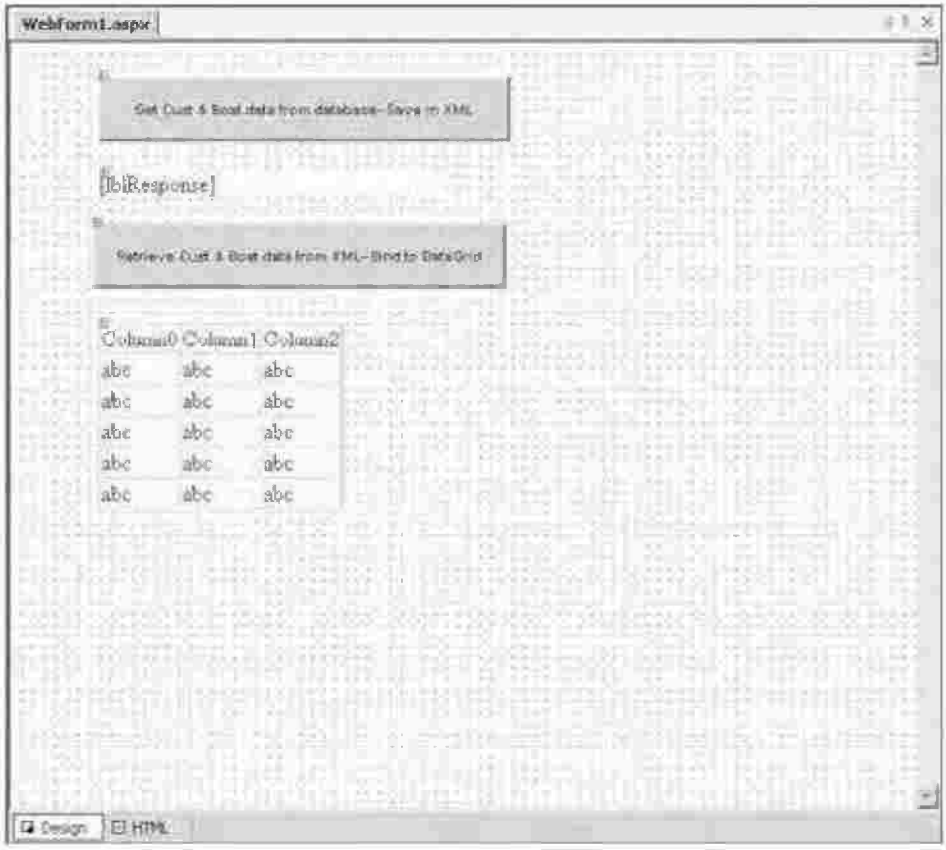
لاحظ على عكس لغة HTML، فإن لغة XML تمثل لغة حساسة لحالة الحرف، وأيضاً إن لغة XML مثل لغة HTML في أنها لغة نصية، ومن ثم يمكن أن ترسل عبر الجدران النارية التي تحجب الأشكال الأخرى.

وأخيراً إن لغة XML (على عكس لغة HTML) تقدم ميزة هامة وهي الربط بين العلامات ومحتواها والتي نطلق عليها "الوصف الذاتي"، فمثلاً بالنظر إلى عبارة <H3> VB .NET Basics </H3>، نجد عدم وجود ربط بين العلامة <H3> والكلمة "VB .NET" وذلك على عكس بيانات المركب المقدمة في الشكل رقم (١٦,٤١).

مثال تطبيق XML
XML Example Application

يوضح هذا المثال كيفية استخدام لغة XML داخل موقع شركة برادشو مارينا، حيث يتم استرجاع بيانات كل من العملاء والمراكب الخاصة بهم من قاعدة البيانات ووضعها داخل كائن من الصنف DataSet، ثم استدعاء الإجراء WriteXML المعرف داخل الصنف XML، وبعد ذلك يقوم البرنامج باسترجاع البيانات من ملف XML وعرضها داخل شبكة بيانات، ويوضح الشكل رقم (١٦,٤٢) نموذج ويب هذا المثال.

وعندما يتم الضغط على الزر العلوي في الشكل رقم (١٦,٤٢)، يقوم إجراء معالجة هذا الزر باسترجاع جميع بيانات العملاء والمراكب من قاعدة البيانات وتخزينها في كائن من الصنف DataSet، ثم يقوم باستدعاء الإجراء WriteXML من هذا الكائن لإنشاء ملف XML (CustomersAndBoats.xml) وتخزينه داخل الفهرس الحالي للموقع، ثم إظهار رسالة نجاح العملية على العنوان الذي يقع أسفل الزر. توضح الأوامر التالية هذا الإجراء حيث يتم أولاً تنفيذ أمر Imports للفضاء المسمى OleDb (للتعامل مع قاعدة البيانات)، ثم تعريف الاتصال بقاعدة البيانات، ثم تنفيذ باقي العمليات لاسترجاع البيانات وتخزينها داخل ملف XML.



الشكل رقم (١٦,٤٢). نموذج ويب مثال لكيفية استخدام لغة XML.

```
Imports System.Data.OleDb
Private Sub btnSaveXML_Click(ByVal sender As System.Object, _
   ByVal e As System.EventArgs) Handles btnSaveXML.Click

   Dim dsCustBoats As New DataSet()
   Dim cnConnection As New OleDb.OleDbConnection _
      ("Provider=Microsoft.Jet.OLEDB.4.0; " & _
       "Data Source=F:\Chap16\CustomerAndBoatDatabase.mdb")

   Dim strSQL As String = "SELECT Name, Address, PhoneNO, " & _
      "StateRegistrationNo, BoatLength, Manufacturer, Year " & _
      "FROM CustomerTable, BoatTable " & _
      "WHERE CustomerTable.PhoneNo = BoatTable.CustomerPhoneno"

   Dim adptBoats As New OleDb.OleDbDataAdapter(strSQL, cnConnection)
   adptBoats.Fill(dsCustBoats, "CustomerAndBoats")
   If dsCustBoats Is Nothing Then
      Exit Sub
   Else
      dsCustBoats.WriteXml(Server.MapPath("CustomerAndBoats.xml"))
      lblResponse.Text = "XML File Created"
   End If
End Sub
```

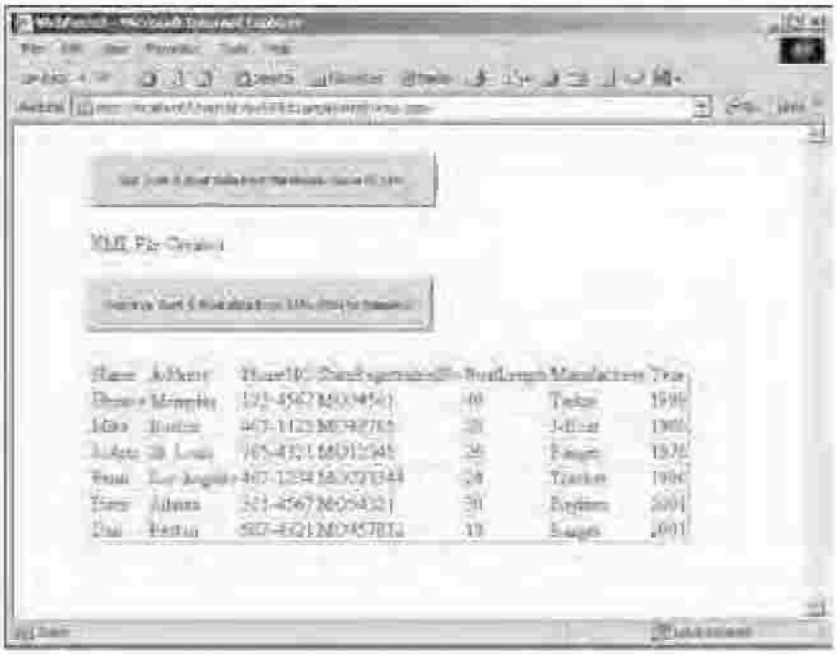
توضح الشفرة التالية جزءاً من ملف XML الذي تم إنشاؤه حيث يظهر بيانات اثنين من العملاء والمراكب التي يملكونها.

```
<?xml version="1.0" standalone="yes"?>
<NewDataSet>
  <CustomerAndBoats>
    <Name>Eleanor</Name>
    <Address>Memphis</Address>
    <PhoneNo>123-4567</PhoneNo>
    <StateRegistrationNo>MO34561</StateRegistrationNo>
    <BoatLength>40</BoatLength>
    <Manufacturer>Tartan</Manufacturer>
    <Year>1998</Year>
  </CustomerAndBoats>
  <CustomerAndBoats>
    <Name>Mike<?Name>
    <Address>Bosten<Address>
    <PhoneNo>467-1122<PhoneNo>
    <StateRegistrationNo>MO98765</StateRegistrationNo>MO98765>
    <BoatLength>28</BoatLength>
    <Manufacturer>J-Boat</Manufacturer>
    <Year>1998</Year>
  </CustomerAndBoats>
```

أما الأوامر التالية توضح إجراء معالجة الزر السفلي في الشكل رقم (١٦,٤٢) والمسؤول عن استرجاع محتويات ملف XML وعرضها داخل شبكة البيانات بالنموذج حيث يتم تعريف كائن من الصنف DataSet واستدعاء الإجراء ReadXML لقراءة ملف XML ومن ثم ربط هذا الكائن بشبكة البيانات.

```
Private Sub btnRetrieveAndBind_Click(ByVal sender As System.Object, _
   ByVal e As System.EventArgs) Handles btnRetrieveAndBind.Click

   Dim dsCustBoats As New DataSet()

   dsCustBoats.ReadXml(Server.MapPath("CustomerAndBoats.xml"))
   DataGrid1.DataSource = dsCustBoats
   DataGrid1.DataBind()

End Sub
```

يوضح الشكل رقم (١٦,٤٣) نموذج الويب بعد الضغط على كل من الزر العلوي والزر السفلي، ومن ثم تخزين ملف XML واسترجاعه. والآن تستطيع أيضاً استخدام لغة XML في تطبيق موقع برادشو مارينا وذلك بتحويل قاعدة البيانات إلى لغة XML ومن ثم تعديل شفرة أصناف التعامل مع البيانات إلى معالجة لغة XML بدلاً من قاعدة البيانات.



الشكل رقم (١٦،٤٣). نموذج الويب بعد الضغط على كل من الزر العلوي والزر السفلي.

تعريف خدمات الويب
Understanding Web Services

يمكن تعريف خدمة الويب تقنياً على أنها طلب يطلب عبر بروتوكول HTTP مثل طلب بحث يرسل لموقع محرك بحث ما أو طلب معلومات عن أسعار منتج من موقع تجارة إلكترونية، ولكن التعريف الأكثر شيوعاً لخدمة الويب هي خدمة معرّفة على كل من شبكة الإنترنت ولغة XML (ولذلك تُدعى خدمة ويب XML) وتؤدي وظيفة محددة. فعلى سبيل المثال إن خدمة Passport لشركة مايكروسوفت تقدم خدمة للمؤسسات المشتركة في خدمة Passport حيث إنها تتحقق من أن المستخدم هو نفس العميل الذي أدخل بيانات لمؤسسة ما، ومن ثم لا يحتاج أن يدخلها مرة أخرى للمؤسسات المشتركة في تلك الخدمة، ومن الأمثلة الأكثر بساطة خدمات الويب التي تقدم خدمة تحويل درجات الحرارة وتحويل العملات.

وصفت شركة مايكروسوفت خدمة ويب XML على أنها برنامج يمكن أن يستدعى عبر شبكة TCP/IP بواسطة تطبيقات أخرى والتي تقوم بإنجاز وظيفة محددة (أي شيء بداية من القيام بحسابات بسيطة والتحقق من صفة بيانات بطاقات الائتمان إلى القيام بمعالجة طلبات معقدة)، ثم تقوم بإعادة النتيجة إلى التطبيق الذي قام باستدعائها عبر الويب حيث يمكن استدعاؤها عبر بروتوكول HTTP أو بروتوكول SOAP، كما يمكنها تبادل البيانات مع البرامج الأخرى باستخدام لغة XML. وأخيراً يمكن أن تتواجد داخل تطبيق ASP.NET لتؤدي مهام الموقع نفسه والتطبيقات الأخرى التي يمكن أن تتعامل عبر الويب، ومن ثم يمكن إنجاز مهام التجارة الإلكترونية بين المؤسسات بسهولة (Business-to-Business).

يقدم إطار عمل .نت أدوات لإنشاء خدمة ويب مثلما يقدم نموذجاً لإنشاء نموذج ويب، مع العلم أن التقنيات التالية ترتبط بإنشاء خدمات الويب:

- بروتوكول SOAP (Simple Object Access Protocol).
- لغة WSDL (Web Service Description Language).
- تقنية UDDI (Universal Description, Integration, and Discovery).

يمثل برتوكول SOAP برتوكولاً للتراسل حيث يقدم وسيلة سهلة لإرسال المعلومات واستقبالها من خدمات الويب حيث يقوم بوضع هذه المعلومات داخل لفافة (Wrapper) التي من شأنها أن تزيل أي مشاكل لها علاقة بشكل البيانات لبروتوكول HTTP وتركيبها، وبذلك فإن وظيفة برتوكول HTTP هي نقل هذه اللفافة من وإلى خدمة الويب (يمكن أن تشتمل اللفافة على كائنات كاملة)، وأحياناً نطلق على اللفافة اسم ظرف (Envelope).

تذكر أن خدمة الويب تشبه الإجراء الذي يستقبل بيانات في شكل معاملات، ثم يعيد نتيجة المعالجة؛ ولذلك تحتاج خدمات الويب إلى طريقة لتوصيف كل من المدخلات والمخرجات للبرامج التي سوف تقوم باستدعائها، ومن ثم نحتاج إلى لغة WSDL التي تقوم بذلك حيث تقوم هذه اللغة بتوصيف تفاعلات خدمات الويب بغض النظر عن بيئة المعالجة التي تنفذ عليها خدمة الويب (Cross-Platform Standard) حيث تقوم لغة WSDL بتوصيف مثلاً أسماء إجراءات خدمات الويب والمعاملات التي تستقبلها والقيم العائدة من تلك الإجراءات.

ولكي تستخدم خدمات الويب، يجب أن يكون هناك برنامج وسيط (Proxy) على جهاز العميل (جهاز التطبيق الذي يستدعي خدمة الويب) حيث يقوم برنامج الوسيط باستقبال طلبات تطبيق العميل ثم لفها (أي وضعها داخل ظرف) مشتملاً على المعاملات (في أشكال لغة XML أيضاً) أو إرسالها إلى البرنامج الوسيط الذي بدوره يستخلص هذه النتيجة وإرسالها إلى تطبيق العميل. تقوم كل من تقنية SOAP وتقنية SWDL بإنجاز هذا كله وإخفائه عن المبرمج الذي يستخدم تقنية ASP.NET.

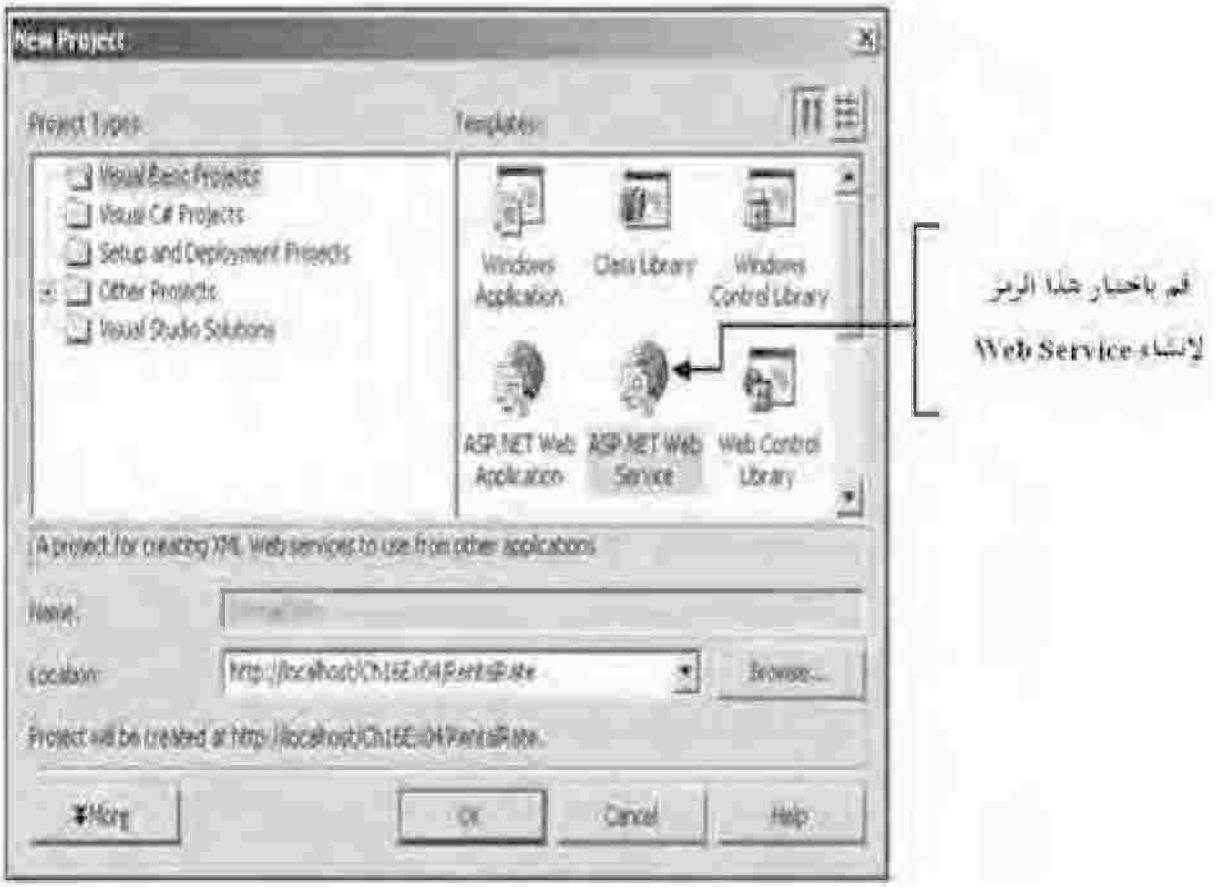
أما تقنية UDDI فإنها تقدم وسيلة سهلة لاكتشاف خدمات الويب المتواجدة عبر الإنترنت حيث قدمت شركة مايكروسوفت خدمة تسمح لمطوري خدمات الويب من تسجيل خدمات الويب الخاصة مجاناً، ومن ثم يمكن للآخرين تصفح قاعدة بيانات خدمات الويب المتواجدة واستخدامها. وعند تطوير خدمة ويب باستخدام تقنية ASP.NET يتم إنشاء ملف وصفي لتلك الخدمة آلياً.

تطوير خدمة ويب لموقع شركة برادشو مارينا
Developing a Web Service for Bradshaw Marina

افترض أن شركة برادشو مارينا أرادت أن تطور خدمة ويب لحساب قيمة إيجار المراسي، حيث يعتمد طريقة حساب الإيجار لخدمة الويب على طول المركب، فإذا كان طول المركب أقل من ٣٠ قدماً سيكون سعر القدم

الواحد يساوي ثلاثة دولارات لأن المركب سيستخدم رصيفاً طبيعياً، أما المراكب ذات الأطوال الأكثر من ٣٠ قدماً ستستخدم رصيفاً خاصاً ومن ثم سيكون سعر القدم الواحدة خمسة دولارات، أما للمراكب ذات الأطوال الأكثر من ٥٠ قدماً فلا يمكن أن ترسي لدى الشركة.

وإذا افترضنا أن جميع شركات إيجار المراسي تستخدم طريقة الحساب نفسها فيمكن استخدام خدمة الويب هذه لجميع الشركات، ولإنشاء خدمة ويب جديدة استخدم نافذة إنشاء مشروع جديد الموضح في الشكل رقم (١٦,٤٤).

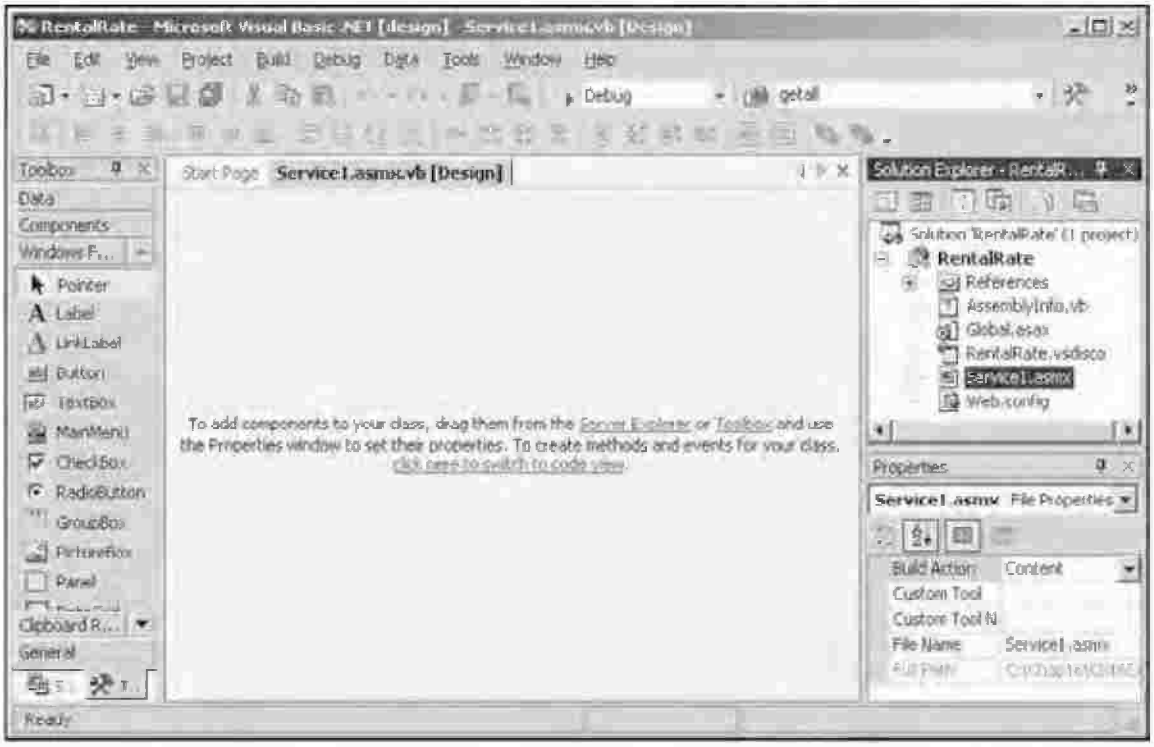


الشكل رقم (١٦,٤٤): استخدام نافذة إنشاء مشروع جديد.

■ لاحظ أنه يمكنك تفحص التطبيق RentalRate الموجود على المجلد Chap16Examples\Chap16Ex04 في ملفات بيانات الطالب.

يوضح الشكل رقم (١٦,٤٥) صفحة البداية لمشروع خدمة الويب RentalRate حيث تنتج لغة فيجوال ستوديو مت جميع الملفات المطلوبة وتستخدم جميع الأصناف المطلوبة لخدمة الويب آلياً. لاحظ أن الاسم الافتراضي للاسم المتد "asmx" هو "Service1" واسم الملف الذي يحوي على الأوامر "Service1.vb"،

ولاحظ أيضاً وجود الملف "RentalRate.vsdisco" داخل نافذة مستكشف الحلول الذي يحتوي على معلومات استكشاف عن خدمة الويب، وبذلك يتم الحصول عليها من قبل التطبيقات الأخرى.



الشكل رقم (١٦,٤٥). صفحة البداية لمشروع خدمة الويب RentalRate.

توضح الأوامر التالية محتويات الملف Service1.asmx حيث إن اللغة المستخدمة هي لغة فيجوال بيسك وإن الملف الذي يحتوي على الإجراءات هو Service1.asmx.vb، وإن اسم صنف خدمة الويب هو "RentalRate.Service1"، ويجب أن تغير اسم الصنف هنا إذا قمت بتغيير اسمه في مكان آخر.

```
<%@ WebService Language="vb" Codebehind="Service1.asmx.vb"
   Class="RentalRate.Service1" %>
```

توضح الأوامر التالية نموذج خدمة الويب الذي يقدمه ASP.NET لملف أوامر خدمة الويب حيث ينفذ أمر Imports للفضاء المسمى System.Web.Service، ثم يضع عنواناً مستعاراً لخدمة الويب الذي يجب أن يتغير من قبل مطوري الخدمة ليحتوي على العنوان الذي يضيف الخدمة، ثم يفضل أن نذكر وصف عمل خدمة الويب (استخدم الرمز = لإسناد كل من عنوان ووصف خدمة الويب).

```
Imports System.Web.Services

<WebService(Namespace:=http://tempuri.org/> _
Public Class Service1
    Inherits System.Web.Services.WebService
```

توضح الأوامر التالية التعديل الذي ستقوم به على الأوامر السابقة.

```
<WebService(Namespace:="http://localhost/Chap16/Chap16Ex04/RentalRate/",
Description :="Determines Dock Rental Rates")> _
Public Class Service1
    Inherits System.Web.Services.WebService
```

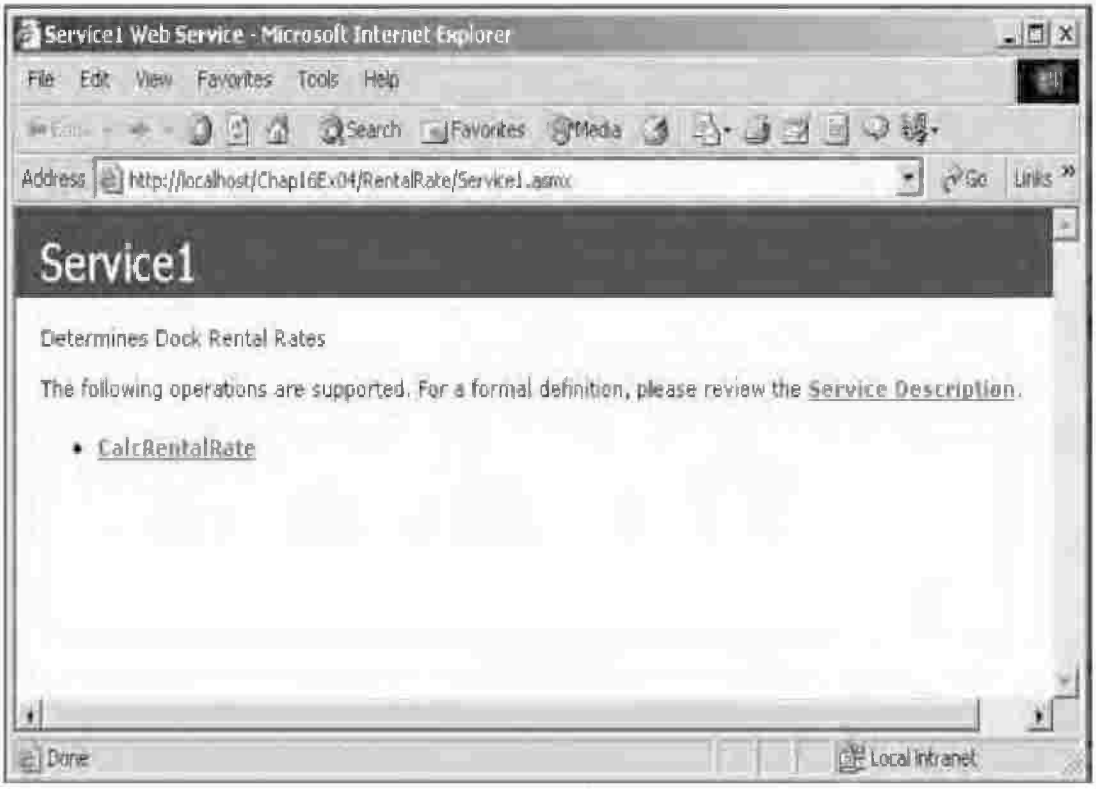
أما بقية أوامر ملف خدمة الويب فإنها توضح مثالاً لإجراء خدمة ويب (انظر الأوامر التالية) لإظهار رسالة الترحيب "Hello World"، فإذا قمنا بإزالة التطبيقات المصاحبة لهذا الإجراء يمكن تنفيذه، وبالنظر للأوامر التالية نجد أن خدمة الويب تشبه إجراءاً طبيعياً ولكن يسبقه الصفة <WebMethod()>.

```
' WEB SERVICE EXAMPLE
' The HelloWorld() example service returns the string Hello World.
' To build, uncomment the following lines, then save and build the project.
' To test this web service, ensure that the .asmx file is the start page
' and press F5.
'
'<WebMethod()> Public Function HelloWorld() As string
'  HelloWorld = "Hello World"
' End Function
```

والآن لنغير الإجراء HelloWorld في الأوامر السابقة ليصبح الإجراء CalcRentalRate المسؤول عن حساب قيمة إيجار المركب طبقاً لطوله التي تستقبله كمعامل. توضح الأوامر التالية الإجراء CalcRentalRate الذي يستقبل المعامل Length، ثم يقوم بحساب قيمة الإيجار بناء على القواعد المذكورة سابقاً.

```
<WebMethod()> Public Function CalcRentalRate(ByVal length As Single) _
        As Decimal

        Select Case length
            Case Is < 0, Is > 50
                Return 0
            Case Is <= 30
                Return length * 3
            Case Else
                Return length * 5
        End Select
    End Function
```

يوضح الشكل رقم (١٦,٤٦) شكل نموذج صفحة الويب "Service1.asmx" عند استدعائها عبر الويب.

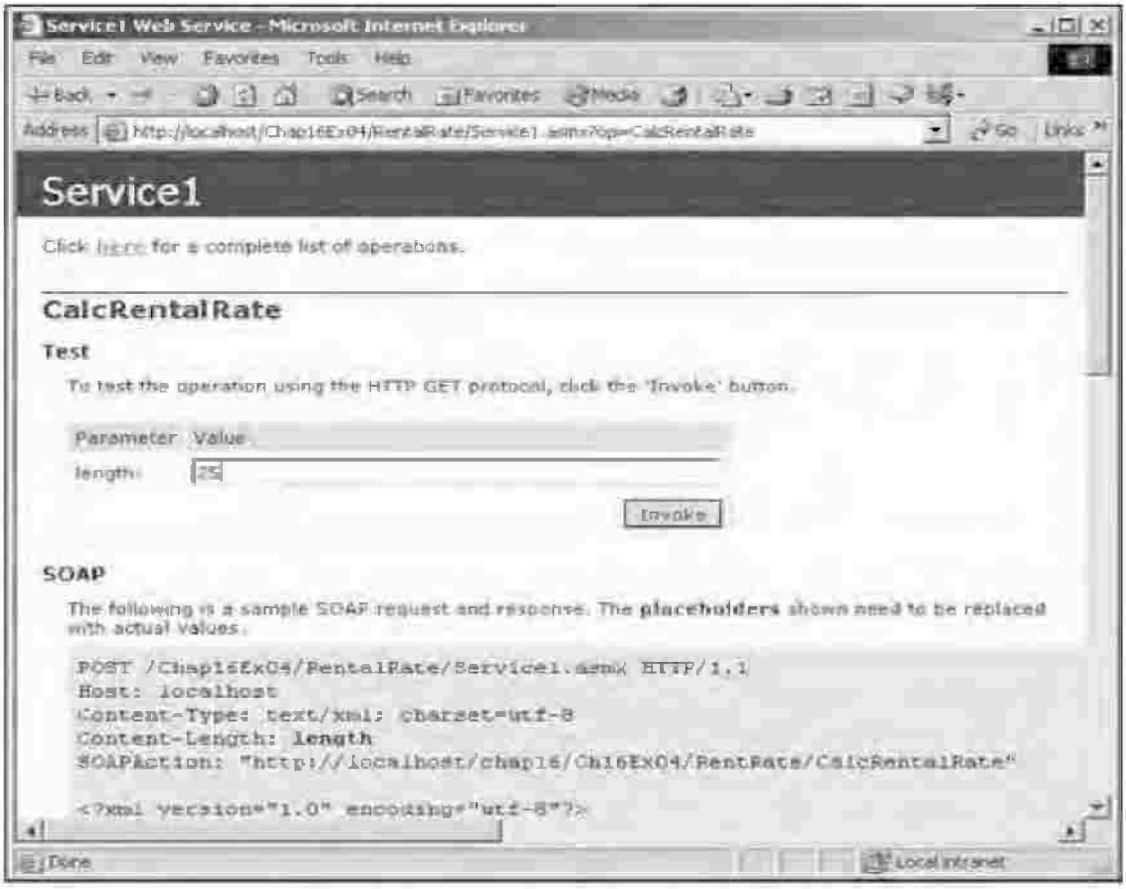


الشكل رقم (١٦,٤٦). نموذج صفحة الويب RentaRate.

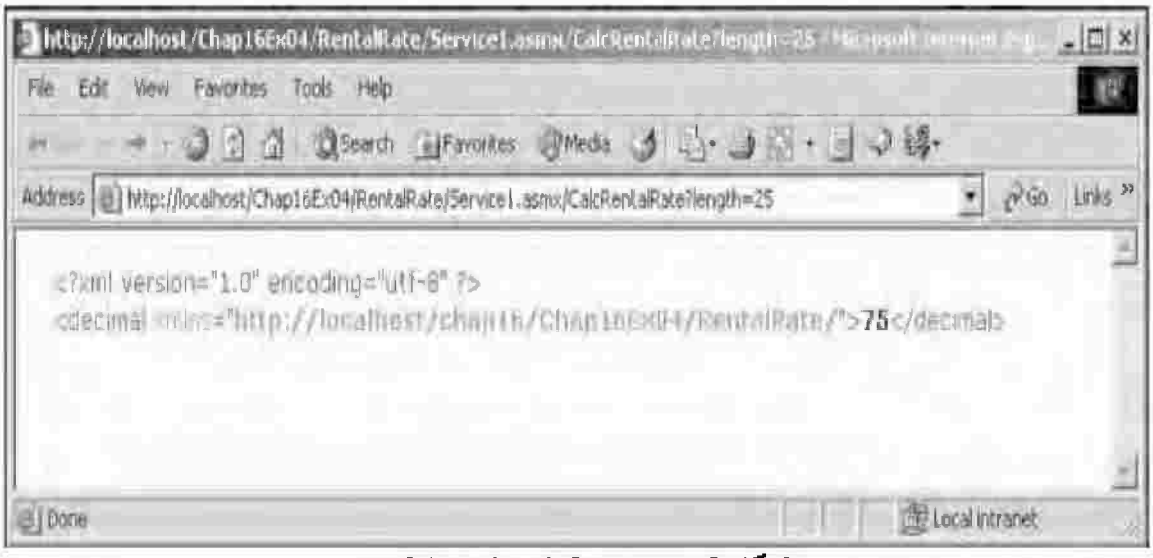
وعند الضغط على الرابط CalcRentalRate الذي يمثل إجراء خدمة الويب ينتج صفحة ويب أخرى (الشكل رقم ١٦,٤٧) الذي يقوم بإنشاء صندوق تلقائياً لاستقبال قيمة معامل الإجراء CalcRentalRate. كما يحتوي النموذج على معلومات حول استخدام خدمة الويب، وزر يقوم بتجربة إجراء خدمة الويب، وفي هذه الحالة تم إدخال القيمة ٢٥ لطول المركب، ولاحظ أيضاً أنه يجب تصفح الصفحة لأسفل لنرى محتويات ملف XML.

وبعد الضغط على زر الشكل رقم (١٦,٤٧) سيظهر النموذج الموضح في الشكل رقم (١٦,٤٨) الذي يوضح قيمة الإيجار ٧٥ دولاراً (لأن طول المركب أقل من ٣٠ قدماً)، ولتحسين شكل المخرجات يمكن استخدام تقنية XSL.

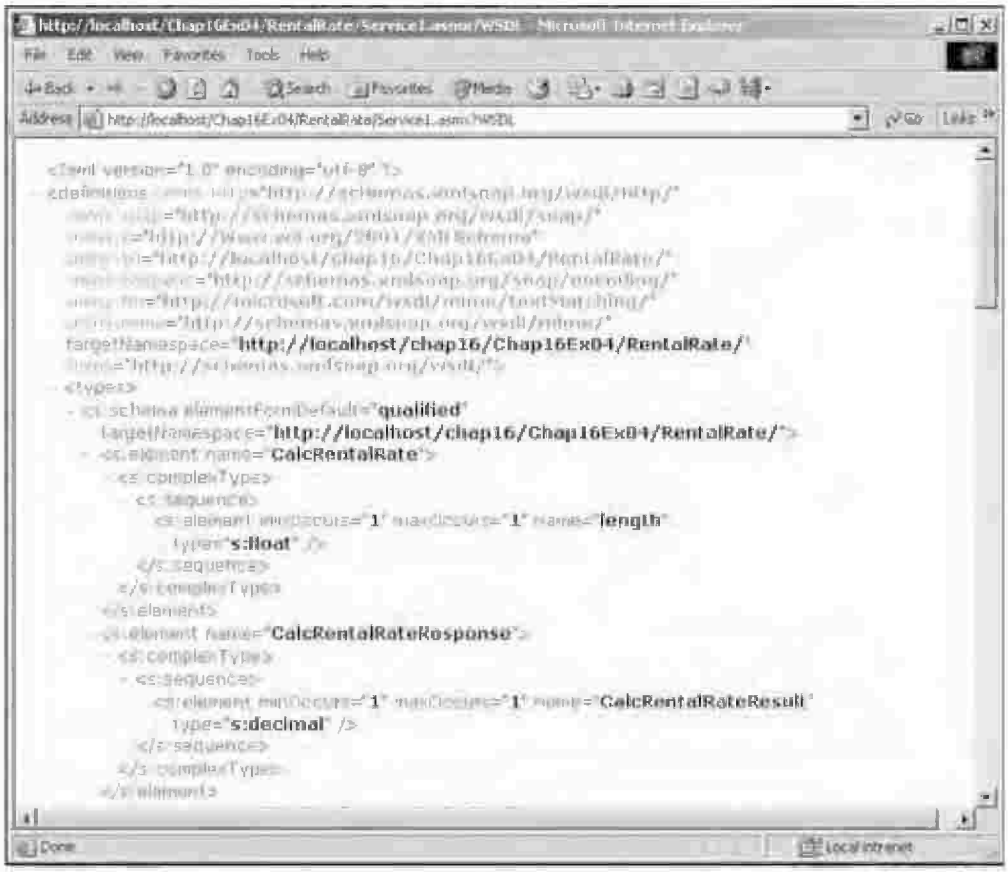
أما إذا قمت بالضغط على الرابط "Service Description" الموضح في الشكل رقم (١٦,٤٦) فسيتم فتح نموذج الويب الموضح في الشكل رقم (١٦,٤٩) الذي تم إنشاؤه تلقائياً باستخدام تقنية WSDL.



الشكل رقم (١٦,٤٧). إجراء خدمة الويب ينتج صفحة ويب أخرى.



الشكل رقم (١٦,٤٨). قيمة نتيجة صفحة الويب.



الشكل رقم (١٦,٤٩). ملف WSDL XML.

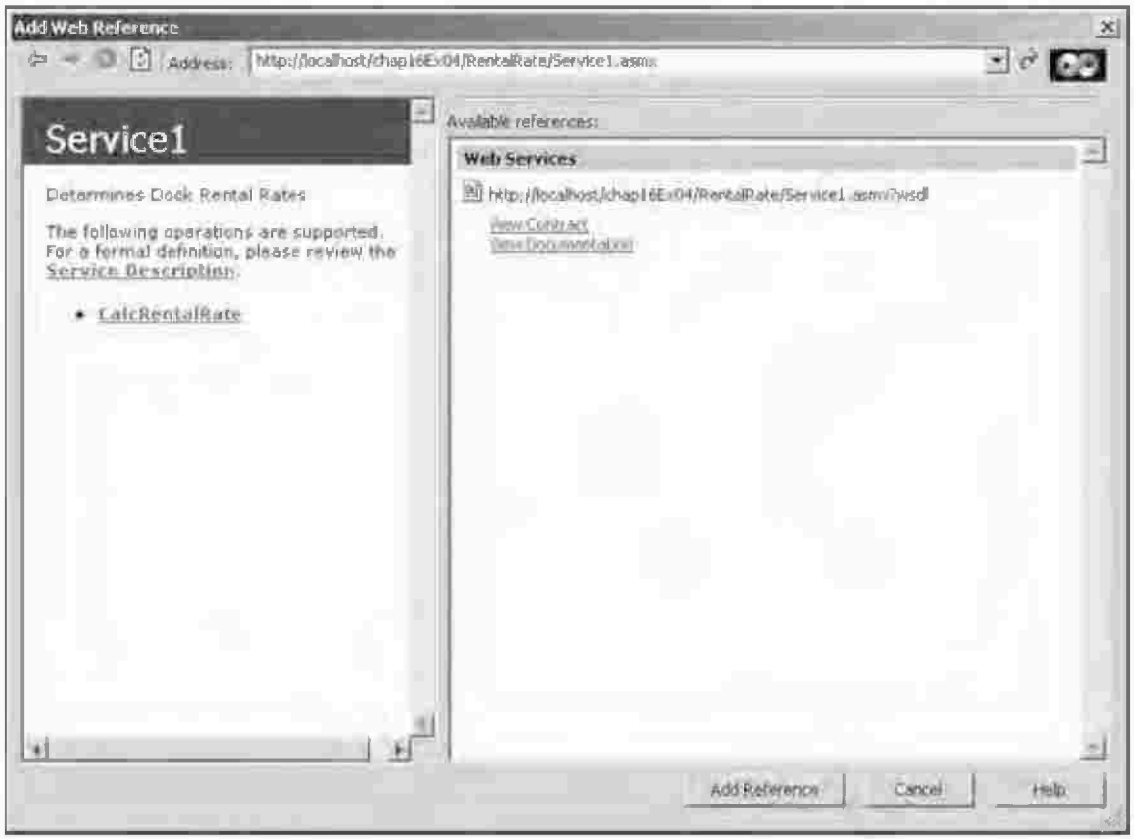
نوضح في المثال التالي تطبيق ASP.NET الذي سيستفيد من خدمة الويب السابقة، وسوف يحتوي نموذج هذا التطبيق على العناصر التالية:

- عنوان يعلم المستخدم معلومات عن خدمة الويب.
- صندوق نص يمكّن المستخدم من إدخال طول المركب.
- عنوان لعرض قيمة الإيجار (النتيجة العائدة من خدمة الويب).
- زر لاستدعاء خدمة الويب.

لاحظ إمكانية أن تعود لهذا التطبيق المتواجد داخل ملفات بيانات الطالب على المجلد: Chap16\Examples\Chap16Ex04.

ولكي يتمكن تطبيق ASP.NET من استخدم خدمة الويب، يجب أولاً أن يتصل بها أو يشير إليها، ولكي نقوم بذلك اضغط بالزر الأيمن على اسم المشروع داخل نافذة مستكشف الحل، ثم اختر "Add Web Reference"، فعندئذ سيظهر الصندوق الحواري الظاهر في الشكل رقم (١٦,٥٠) الذي يحتوي على مواقع UDDI والتي تمكّنك

من البحث على خدمة الويب التي تريدها، كما تحتوي هذه النافذة على صندوق عنوان الذي يستخدم لكتابة عنوان خدمة الويب مباشرة عند معرفته، وبما أننا نعلم عنوان خدمة الويب الذي قمنا بتطويرها سابقاً نقوم بكتابة عنوانها مباشرة، وسنلاحظ أن اسم خدمة الويب "CalcRentalRate" يظهر داخل الجزء الأيمن من النافذة (تذكر أن الاسم هو نفس اسم الإجراء المعرف داخل خدمة الويب)، وأيضاً ستظهر روابط أخرى لخدمة الويب داخل الجزء الأيمن من النافذة. ولإتمام عملية الاتصال قم بالضغط على الزر Add Reference. والآن تمت عملية الإشارة إلى خدمة الويب داخل نافذة مستكشف الحل.

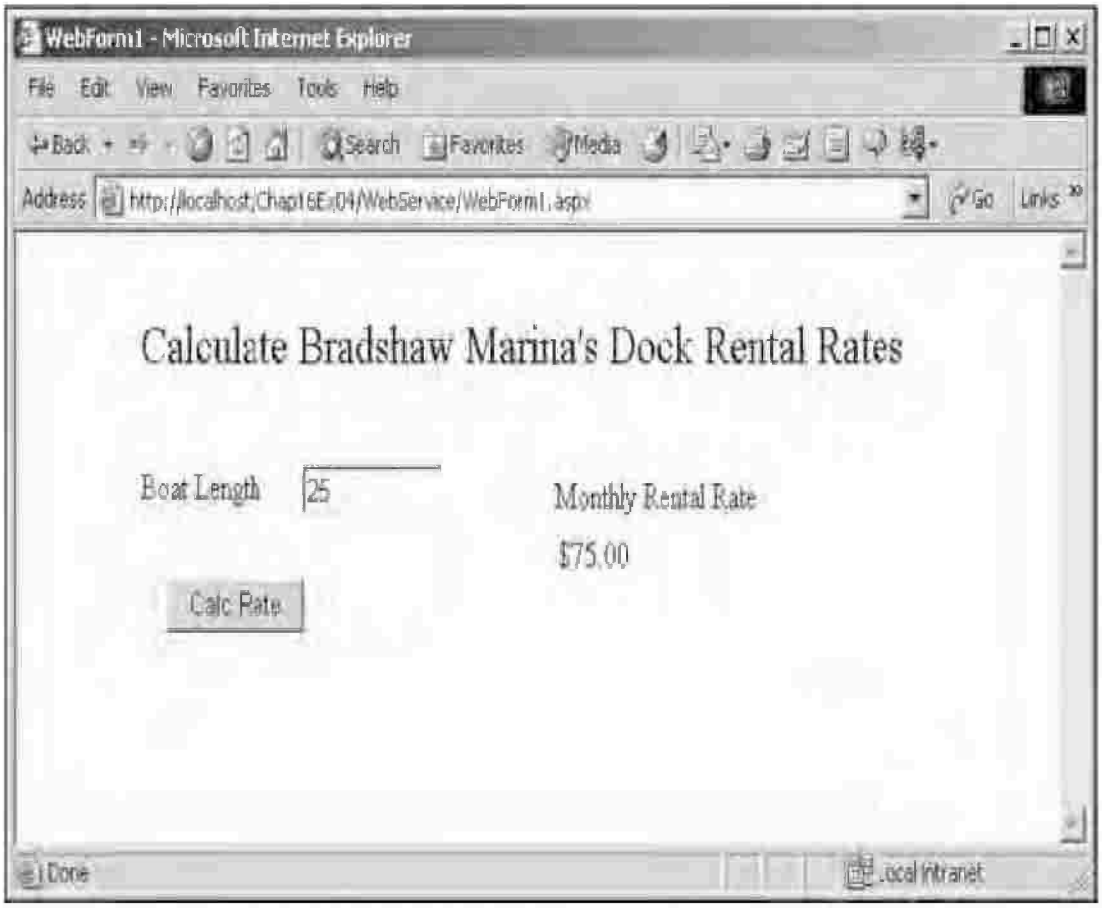


الشكل رقم (١٦,٥٠). نافذة استرجاع إضافة ويب.

توضح الأوامر التالية إجراء معالجة الزر "Calc Rate" المتواجد داخل نموذج الشكل رقم (١٦,٥١) حيث يقوم أولاً بتعريف الكائن CSS من خدمة الويب، ثم يقوم بتعريف متغير لطول المركب، ثم يقوم باستدعاء الإجراء CalcRentaRate من الكائن CSS، ثم عرض النتيجة العائدة داخل العنوان lblRentaRate وذلك بعد تحويلها إلى شكل العملة.

```
Private Sub btnCalcRate_Click(ByVal sender As System.object,_
   ByVal e As System.EventArgs) Handles btnCalcRate.Click

   ' Declare an instance of the RentalRate Web service
   Dim orr As New WebService.localhost.Service1()
   ' Declare a variable to hold the boat length
   Dim boatLength As Single
   boatLength=CType(txtBoatLength.Text,Double)
   lblRenalRate.Text=Format(orr.CalcRentalRate(boatLength), "Currency")

End Sub
```

يوضح نموذج الويب الظاهر في الشكل رقم (١٦,٥١) استدعاء خدمة الويب داخل تطبيق ASP.NET حيث كان الغرض من هذا المثال هو عرض كيفية إنشاء خدمة ويب واستخدامها داخل تطبيق ويب ASP.NET.



الشكل رقم (١٦,٥١). نموذج ويب يستخدم خادم الويب RentalRate.

ملخص الفصل

Chapter Summary

- تعتمد خدمة الويب على بروتوكول HTTP المعروف بعدم تذكره لحالة المستخدم (Stateless Protocol)، ولذلك تقدم تقنية ASP التقليدية الاحتفاظ بحالة العميل عن طريق استخدام الكائن Application والكائن Session وملفات Cookies.
- تعرف متغيرات التطبيق داخل الكائن Application لتكون مشتركة على مستوى جميع متصفحي الموقع. وتعرف متغيرات الجلسة داخل الكائن Session لتكون خاصة بكل مستخدم على حده طوال فترة تصفحه لصفحات الموقع. أما ملفات Cookies فتستخدم للاحتفاظ بمعلومات قليلة (أقل من ٤ كيلو بايت) داخل ملفات على جهاز المستخدم التي تبقى صالحة لفترة زمنية محددة.
- تعتمد برامج معالجة النصوص سواء كانت مكونة من طبقتين أو من ثلاث طبقات على نموذج طلب/استجابة.
- تقدم تقنية ASP.NET تقنية "عنصر مستخدم" (User Control) وهو عبارة عن نموذج ويب يؤدي إلى توافق نماذج الويب. كما تقدم تقنية ASP.NET أسلوب ربط البيانات (Data Binding) الذي يعد وسيلة سهلة لارتباط خاصية عنصر ما بمصدر للبيانات.
- يقدم العنصر DataGrid تقنية مرنة وقوية لعرض المعلومات على نماذج الويب.
- تقدم تقنية ASP.NET عناصر التحقق من صحة البيانات (Validation Controls) للتحقق من صحة البيانات المدخلة.
- فصل العميل عن الخادم هو سبب عدم سماحية الاستجابة لجميع الأحداث من قبل عناصر خادم الويب، فعلى سبيل المثال إن الحدث mouse-over الذي ربما يقابل بتغيير لون النص يتطلب العودة إلى الخادم ثم الرجوع لعميل الويب؛ لذا يتم تجميع البيانات كلها وإرسالها مرة واحدة عند حدوث أحداث مميزة مثل الضغط على أي زر.
- تستطيع تقنية ASP.NET الفصل بين استدعاء الصفحة للمرة الأولى أو لمرة متكررة بما يسمح تنفيذ متطلبات استدعاء الصفحة للمرة الأولى.
- يتم ترجمة صفحات الويب إلى لغة الآلة لتنفيذ أسرع عند استدعاء الصفحة باستخدام مترجم من النوع just-in-time.
- إن لغة XML تقدم وسيلة فعالة للتكامل بين مصادر بيانات مختلفة سواء داخل مؤسسة واحدة أو بين عدة تطبيقات لمؤسسات مختلفة عبر الإنترنت.

- تمتلك تقنية ASP.NET على إجراءات XML التي تؤدي إلى تخزين بيانات من النوع XML واسترجاعها ومعالجتها.
- يمكن تعريف خدمة الويب تقنياً على أنها طلب يُطلب عبر بروتوكول HTTP مثل طلب بحث يرسل لموقع محرك بحث ما أو طلب معلومات عن أسعار منتج من موقع تجارة إلكترونية.

المصطلحات الأساسية

Key Terms

تقنية ASP	عنصر مستخدم (User Control)
الكائن Application	ربط البيانات (Data Binding)
الكائن Session	عنصر DataGrid
ملفات Cookies	صحة البيانات (Validation Controls)
تقنية ASP.NET	لغة XML
الصفة Postback	حالة State
ربط البيانات Data Bound	خدمة الويب Web Service
لغة توصيف خدمة الويب UDDI	

أسئلة المراجعة

Review Questions

١- هل تظن أن لغة فيجوال بيسك .نت ملائمة لتطوير تطبيقات الويب؟

٢- لماذا يوجد بعي الأحداث مثل OnBlur لا تدعم من قبل عناصر تحكم خادم الويب؟

٣- ما هو بروتوكول اللاحاجة؟ هل تستطيع توضيحه في المصطلحات اليومية؟

٤- ما هي مميزات ملف الكود-الخلفي؟

٥- إذا كنت تريد الاحتفاظ ببعض المعلومات عن زيارة عميل لموقع إنترنت، ما هي الوسائل المتاحة لإتمام هذه المهمة؟

٦- ماذا يعني تعبير "data bind"؟

٧- ما هي مميزات شبكة البيانات؟

٨- إذا كنت تحتاج أن يكون لديك صفحات ويب ثابتة، ما هي الأعمال التي تتطلب منك في لغة فيجوال بيسك .نت؟

٩- في عنصر تحكم المستخدم، هل تستطيع استخدام كل من HTML وعنصر تحكم خادم الويب؟ إذا كان الأمر كذلك، ما هي إرشادات هذا القرار؟

١٠- ما هي العملية التي تستخدمها إذا كنت تريد كتابة شفرة كارت تسوق؟

١١- ماذا يعني تعبير "postback"؟ إلى أين تقرر في لغة فيجوال بيسك .نت حدث postback؟

١٢- ما هي عناصر التحكم المستخدمة في تغليف البيانات؟

أسئلة المناقشة

Discussion Questions

١- هل تعتقد أنه يمكن استخدام عناصر ASP.NET التحقق من صحة البيانات بشكل دائم؟

٢- هل تعتقد أن الاهتمام الزائد بأمان الحاسب أدى كثيراً من الناس إلى القيام بإيقاف ملفات cookies داخل صفحات الويب؟

٣- اذكر مميزات تقنية ASP.NET عن تقنية ASP التقليدية؟ وهل يوجد من التطبيقات التي تتطلب استخدام تقنية ASP التقليدية حتى في وجود تقنية ASP.NET؟

٤- ما هي مميزات استخدام متغيرات الكائن Application وعيوبه؟

٥- ما هي المواصفات التي تريد أن تضيفها إلى موقع شركة برادشو مارينا؟

مشاريع الفصل

Projects

١- قم بتطوير موقع شخصي لك باستخدام تقنية ASP.NET، ولكن تأكد من إضافة User Control، كما يجب أن تكون مبدعاً في تحديد البيانات التي يجب تخزينها داخل قاعدة البيانات.

٢- قم بتطوير موقع حجز مواعيد لعيادة خاصة باستخدام تقنية ASP.NET، ولكن تأكد من إضافة User Control، كما يجب أن تكون مبدعاً في تحديد البيانات التي يجب تخزينها داخل قاعدة البيانات.