الفصل الثاني عشر

نماذج الويب، ولغة HTML، وتقنية ASP.NET

WEB FORMS, HTML, AND ASP.NET

أهداف الفصل:

- مراجعة مبادئ الإنترنت والويب.
- مراجعة مبادئ لغة HTML.
- استخدام نماذج لغة HTML.
- استخدام تقنية ASP.NET لتطوير صفحات الويب.
- إنشاء تطبيق ويب باستخدام تقنية ASP.NET.
- إنشاء تطبيق شامل على الويب.

تعرف الإنترنت بنظام شبكي هائل يتكون من عدد كبير جداً من شبكات الحاسب الآلي على مستوى العالم من خلال برتوكول يسمى Transmission Control Protocol/Internet Protocol ويختصر باسم TCP/IP، ويقدر عدد مستخدمي الإنترنت حالياً حوالي ٥٠٠ مليون مستخدم ويوجد أكثر من ١٠٠ مليون موقع إنترنت، ولأن خدمة الويب تتميز بواجهة جذابة حيث تحتوي على رسومات وعلى عناصر متحركة فإنها تمثل أهم خدمة تقدم على شبكة الإنترنت، ولقد أثرت خدمة الويب في نظم المعلومات لأنها لا تهتم بالبيئة التي سيعمل عليها النظام حيث إنها تستخدم برتوكولاً مثالياً يسمى Hypertext Transmission Protocol (ويختصر بالاسم HTTP) لمعالجة المعلومات ولغة مثالية تسمى Hypertext Markup Language (وتختصر بالاسم HTML) لعرض المعلومات على شاشة الحاسب الآلي، ومن ثم يحتاج المستخدم متصفح ويب فقط لكي يستخلص المعلومات ويعرضها على الشاشة بغض النظر عن البيئة الحالية (نظام تشغيل ويندوز أو ماكنتوش أو غيرهما).

obeikandl.com

يقدم هذا الفصل تطوير تطبيقات الويب باستخدام إطار عمل مايكروسوفت .نت، وبعد الانتهاء من هذا الفصل سوف تحصل على المفاهيم الأساسية لتطوير تطبيقات الويب والمقدرة على تطوير كل من المواقع ثابتة المحتوى (Static Web Sites) والمواقع متغيرة المحتوى (Dynamic Web Sites) باستخدام تقنية ASP.NET. سوف نطور تطبيقات الويب باستخدام أسلوب تطوير تطبيقات الويندوز نفسه وذلك بتصميم نماذج الويب باستخدام عناصر رسومية مثل الصناديق النصية والعناوين والأزرار. ولإنشاء تطبيق ويب نقوم باختيار أمر "New Project"، ثم نختار "Visual Basic Projects" من قائمة أنواع المشاريع، ثم نختار "ASP.NET Web Applications"، مع العلم أن اللغة المستخدمة لتطوير أمثلة تطبيقات الويب في هذا الفصل هي لغة فيجوال بيسك .نت.

سوف يبدأ هذا الفصل بتقديم مفاهيم الإنترنت ولغة HTML وتقنية ASP.NET، فإذا كان لديك خبرة بهذه المواضيع فيمكنك عزيزي القارئ قراءتها سريعاً أو الانتقال مباشرة إلى الموضوع الذي يليه.

الإنترنت وأساسيات الويب

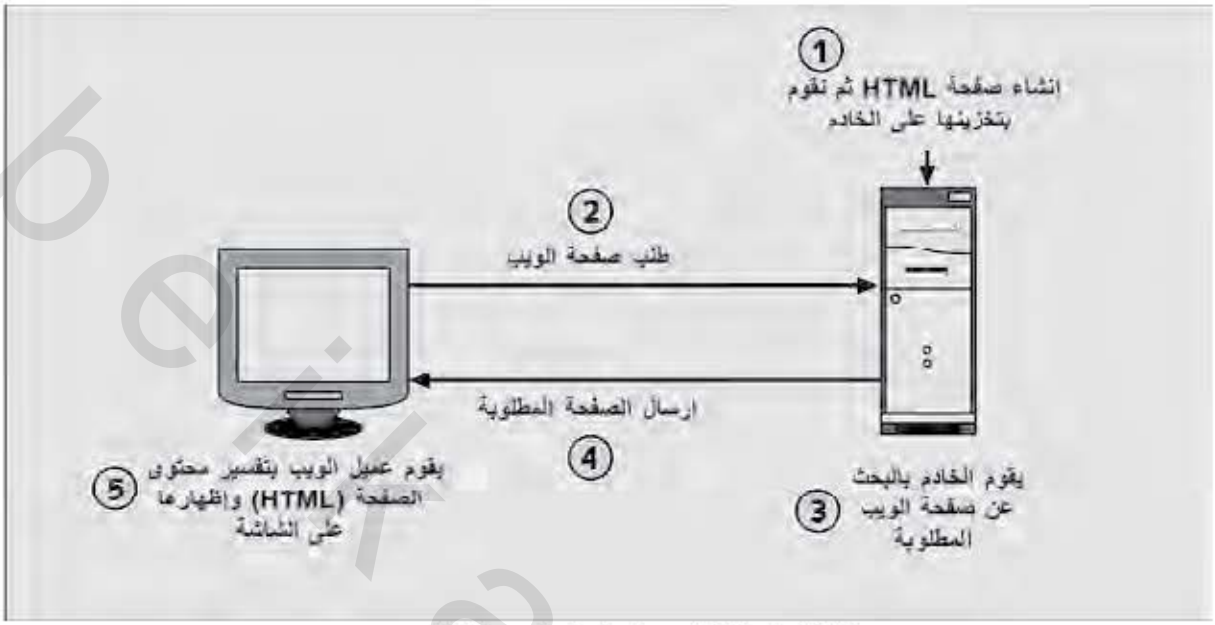
Understanding Internet and Web Fundamentals

تعتمد خدمة الويب على بنية الخادم/العميل (Client/Server) الذي يمثل أحد أشكال المعالجة الموزعة حيث تنقسم مهام البرنامج إلى قسمين وهما: قسم ينفذ لدى الخادم، وقسم ينفذ لدى العميل. فعلى سبيل المثال، عندما يطلب المستخدم صفحة ويب من خلال جهاز عميل الويب، فإن خادم الويب يبحث عن هذه الصفحة ثم يرسلها له، وعندئذ يقوم عميل الويب بترجمة لغة HTML الموجودة داخل هذه الصفحة، ثم يعرضها أمام المستخدم. يتم الاتصال بخادم الويب من خلال عنوان الويب الخاص به الذي يسمى Uniform Resource Locator ويختصر بالاسم URL، ولذلك عندما نريد أن نزور موقعاً معيناً نكتب العنوان الخاص به داخل صندوق النص الخاص ببرنامج عميل الويب (متصفح الويب) مثل برنامج مايكروسوفت إنترنت إكسبلورر. تخزن كل صفحة ويب في الملف الذي يأخذ اسماً ممتداً هو ".htm" أو ".html"، ويحتوي هذا الملف على لغة HTML (ربما يحتوي الملف على شفرة أخرى غير لغة HTML).

صفحات ويب ثابتة المحتوى وأخرى متغيرة المحتوى

Static and Dynamic Web Pages

تحتوي صفحات الويب ثابتة المحتوى على لغة HTML التي تستخدم لعرض المعلومات. فعلى سبيل المثال، يمكن لمؤسسة ما أن تعرض سياستها وإجراءات عملها داخل صفحة ويب ثابتة المحتوى، وعندما يطلب المستخدم زيارة هذه الصفحة ترسل له كما هي دون تغيير من الخادم إلى برنامج العميل الذي يفسر ويعرض محتوياتها، ويوضح الشكل رقم (١٢,١) كيفية معالجة صفحات الويب ثابتة المحتوى.



الشكل رقم (١٢,١). معالجة صفحة ويب ثابتة المحتوى.

يوضح هذا الشكل خمس خطوات لمعالجة صفحات الويب ثابتة المحتوى :

١- إنشاء صفحات HTML وتصميمها، ثم تخزينها على خادم الويب ضمن موقع ويب ما.

٢- يطلب المستخدم زيارة هذه الصفحة.

٣- يبحث خادم الويب عن الصفحة المطلوبة.

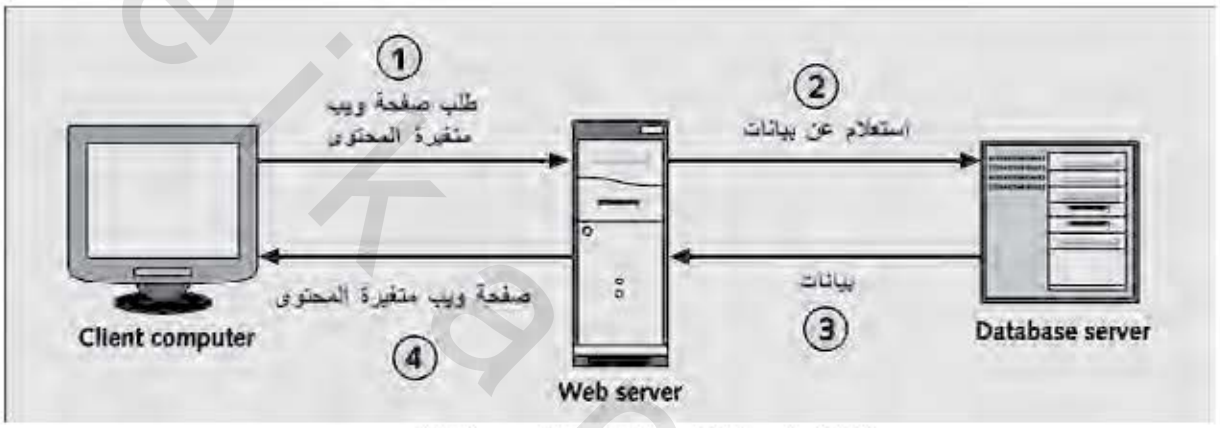
٤- يرسل خادم الويب الصفحة المطلوبة إلى عميل الويب.

٥- يترجم عميل الويب لغة HTML المخزنة داخل الصفحة وعرضها على شاشة المستخدم.

وعلى الجانب الآخر، إن صفحات الويب متغيرة المحتوى تحتوي على محتويات تتغير بناء على طلبات المستخدم ورغباته. فعلى سبيل المثال، إن مواقع الويب للتسوق عبر الإنترنت (مثل موقع www.amazon.com) تستخدم صفحات متغيرة المحتوى، وعندما يبحث المستخدم عن منتج محدد فإن معلومات هذا المنتج تأتي من قاعدة بيانات الموقع، ومن ثم تتغير محتويات الصفحة بمعلومات هذا المنتج.

ولكي ننشئ صفحات ويب متغيرة المحتوى، نحتاج أن نكتب برامج تعمل لدى عميل الويب (Client-Side) أو برامج تعمل لدى خادم الويب (Server-Side). لقد طورت شركة مايكروسوفت تقنية لتطوير صفحات الويب متغيرة المحتوى على الخادم تسمى Active Server Pages وتختصر بالاسم ASP، ثم أطلقت الإصدار ASP.NET مع إصدار فيجوال ستوديو نت، وبذلك أطلقت على تقنية ASP الأولى اسم "Classic ASP". تسمح تقنية ASP إضافة برامج نصية (Scripts) داخل صفحات الويب التي تنفذ لدى الخادم، ومن ثم تنتج صفحات متغيرة المحتوى، ولكي تطور هذه البرامج

يتطلب الأمر معرفة لغة برمجة نصية مثل لغة VBScript أو لغة JavaScript. يوجد هناك العديد من تقنيات تطوير صفحات الويب متغيرة المحتوى لدى الخادم مثل تقنية ASP، وتقنية ASP.NET، وتقنية CGI، وتقنية JSP، وتقنية Cold Fusion، وتقنية PHP. تتصل نظم تطبيقات الويب بقاعدة بيانات أو أي مصدر آخر للبيانات، ويوضح الشكل رقم (١٢,٢) خطوات طلب صفحة ويب متغيرة المحتوى التي تعتمد على وجود قاعدة بيانات متصلة بخادم الويب.



الشكل رقم (١٢,٢). معالجة صفحة ويب متغيرة المحتوى.

توضح الخطوات التالية خطوات معالجة صفحات متغيرة المحتوى:

١- يطلب المستخدم صفحة ويب متغيرة المحتوى مثل طلبه سابقاً لصفحة ويب ثابتة المحتوى، ولكن يمكنه هنا إرسال بيانات إلى الخادم من خلال نموذج ويب.

٢- يطلب خادم الويب من خادم قاعدة البيانات معلومات مطلوبة وذلك بناء على معالجة صفحة الويب على نماذج الويب.

٣- يستخلص خادم قاعدة البيانات المعلومات المطلوبة من قاعدة البيانات ثم يعيدها إلى خادم الويب.

يعالج خادم الويب المعلومات المطلوبة وتحويلها إلى لغة HTML لإرسالها إلى عميل الويب الذي يقوم بعرضها إلى المستخدم.

إعداد برنامج خادم الويب (IIS)
Setting Up an IIS Server

عندما تستخدم تقنية مايكروسوفت لتطوير تطبيقات الويب، يجب عليك أن تستخدم برنامج خادم الويب الخاص بذلك والذي يسمى Internet Information Services ويختصر بالاسم IIS حيث يجب تنصيب هذا البرنامج على جهاز منفصل عن جهاز الخادم، ولكن أثناء عملية التطوير يمكن تنصيبه على نفس جهاز التطوير، ومن ثم

سوف يكون لديك على الجهاز كل من عميل الويب وخادم الويب على نفس الجهاز. إن جميع تطبيقات الويب التي سوف تطورها في هذا الكتاب ستعمل بهذا السيناريو، أما إذا كان برنامج خادم الويب IIS يوجد لديك على جهاز منفصل على الشبكة، فعليك سؤال مدير الشبكة عن كيفية الاتصال بهذا الخادم.

يأتي برنامج IIS مع نظام التشغيل ويندوز ٢٠٠٠ وما بعده، ولتثبيت هذا البرنامج على جهاز الحاسب الآن، فإن الخطوات التالية سوف تقودك لكي تقوم بتنصيبه ومن ثم يمكنك استخدام الجهاز كخادم ويب، ولأن خطوات تنصيب برنامج IIS على كل من ويندوز ٢٠٠٠ وويندوز XP تتشابه، فقد تم تجميع الخطوات سوياً (يجب أن تعرف أولاً هل البرنامج يوجد بالفعل على الجهاز أم لا). لكي تنصب برنامج IIS على ويندوز ٢٠٠٠ أو ويندوز XP اتبع الآتي:

١- افتح نافذة لوحة التحكم (Control Panel): اختر Start، ثم Settings، ثم "Control Panel" في حال استخدامك ويندوز ٢٠٠٠، أو اختر Start، ثم "Control Panel" في حال استخدامك ويندوز XP.

٢- اضغط ضغطاً مزدوجاً على رمز "Add/Remove Programs"، فعندئذ ستظهر نافذة بعنوان "Add or Remove Programs" كما هو واضح في الشكل رقم (١٢,٣) والتي توضح البرامج المتواجدة حالياً على جهاز الحاسب الآلي الخاص بك.

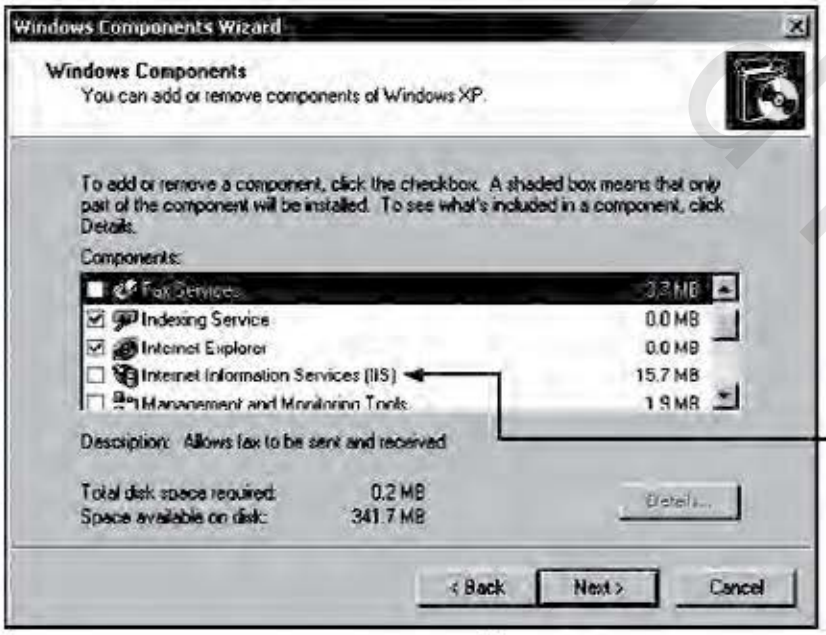
٣- اختر من القائمة اليسرى الخيار "Add/Remove Windows Components"، فعندئذ ستظهر نافذة معالجة مكونات الويندوز موضحةً جميع مكونات الويندوز التي يمكن تنصيبها أو إلغاؤها كما هو واضح في الشكل رقم (١٢,٤). ابحث عن المكون "Internet Information Services (IIS)"، فإذا كان هذا المكون مؤشراً عليه بالفعل فهذا يعني أن برنامج IIS منصب على جهازك الآن وكل ما يجب عليك فعله الآن هو اختياره فقط من القائمة (كن حذراً حيث يجب أن لا تلغي التأشير)، ثم انتقل مباشرة إلى الخطوة التالية. أما إذا كان المكون غير مؤشر عليه فعليك أن تؤشر عليه ثم تضغط على الزر NEXT، فعندئذ ربما يطلب منك إدخال قرص ويندوز ٢٠٠٠ (أو ويندوز XP) داخل محرك الأقراص.

٤- قم بالضغط على الزر Details أثناء اختيار برنامج IIS داخل نافذة معالج مكونات الويندوز، فعندئذ ستظهر نافذة برنامج IIS موضحةً المكونات الفرعية للبرنامج والمنصبة حالياً على جهازك كما هو واضح في الشكل رقم (١٢,٥).

٥- اختر المكونات الفرعية: "World Wide Web Service"، و "FrontPage 2000 Server Extensions"، و"Internet Information Services Snap-In"، وإذا كان برنامج IIS يأخذ الإصدار ٥,٠ والموزع مع ويندوز ٢٠٠٠، فيجب عليك أن تختار "Script Debugger" ثم اضغط على الزر OK.

٦- اضغط على الزر Next ثم على الزر Finish لإكمال عملية تنصيب برنامج IIS.

الشكل رقم (١٢,٣). لائحة إضافة البرامج وفحصها.



الشكل رقم (١٢,٤). نموذج معالج تنزيل مكونات الويندوز.

الشكل رقم (١٢,٥). مكونات وخدمات برنامج IIS.

يستخدم برنامج IIS المجلد "C:\inetpub\wwwroot" كمجلد افتراضي لنشر مواقع الويب، وسوف نتحقق في الخطوات التالية من صحة إنشاء هذا المجلد ومن صحة عمل البرنامج. يأخذ خادم برنامج IIS نفس اسم جهاز الحاسب الخاص بك، وسوف نستخدم الاسم "localhost" لخادم الويب وهو الاسم الافتراضي لأي خادم يعمل مع العميل على نفس الجهاز.

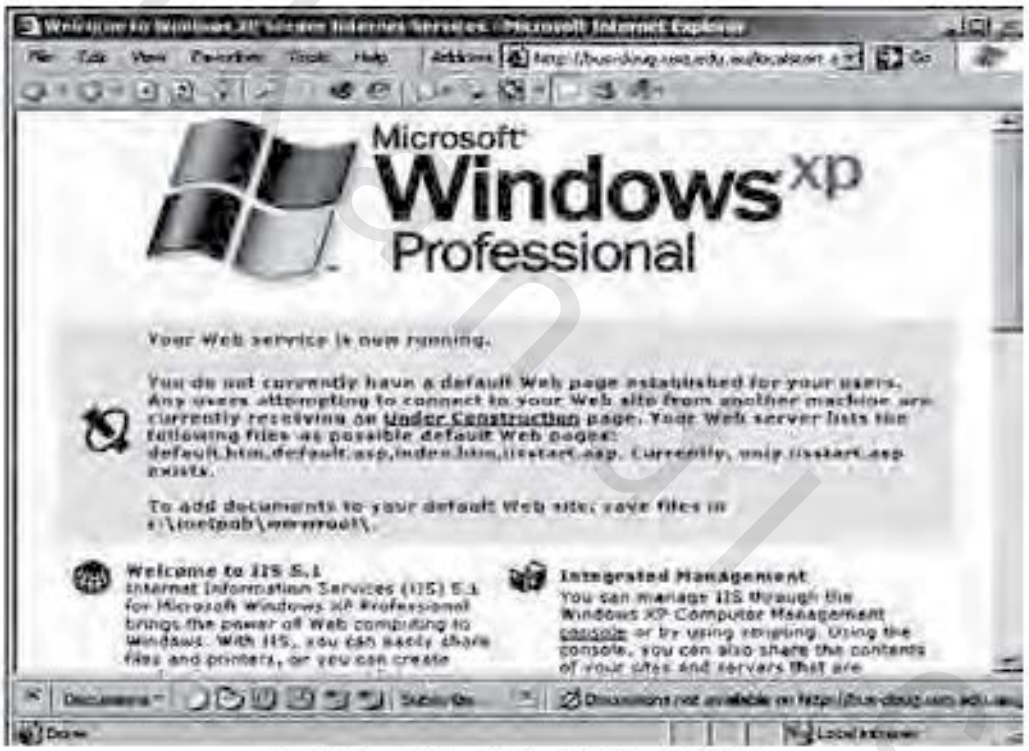
لكي تختبر عملية تنصيب برنامج IIS اتبع الخطوات الآتية:

١- افتح برنامج "مستكشف الويندوز"، ثم تصفح الفهرس "C:\inetpub\wwwroot"، ثم ابحث عن الملف "localstart.asp" الذي يمثل صفحة اختبار.

٢- ولمعرفة اسم جهاز الحاسب في ويندوز ٢٠٠٠، اختر Start، ثم Settings، ثم "Control Panel"، ثم اضغط ضغطاً مزدوجاً على "Network and Dial-up Connections"، ثم اختر Advanced في القائمة الرئيسة، ثم اختر "Network Identification"، ثم اختر Properties لكي ترى اسم جهازك الحاسب (كما يمكنك أيضاً تغيير هذا الاسم من نفس النافذة). ولكي تعرف اسم جهازك في ويندوز XP، اضغط على Start، ثم اضغط على "Control Panel"، ثم اضغط ضغطاً مزدوجاً على System لكي تفتح نافذة "System Properties"، ثم اضغط على مسطرة "Computer Name"، ثم اضغط على زر Change لكي ترى اسم الجهاز (يمكنك تغيير اسم الجهاز من نفس النافذة). سوف تستخدم اسم الجهاز في الخطوة التالية، والآن أغلق جميع النوافذ المفتوحة.

٣- لكي تتصفح صفحات الويب الموجودة على خادم الويب، يجب عليك أن تشغل برنامج "Internet Explorer" (عميل الويب)، ويمكنك تشغيل هذا البرنامج بالضغط المزدوج على رمز "Internet Explorer" الموجود على سطح المكتب أو اختر Start ثم "Internet Explorer".

٤- اكتب "http://your_computer_name/localstart.asp" داخل صندوق نص العنوان حيث يمثل "your_computer_name" اسم جهاز الحاسب الخاص بك والذي حصلت عليه من الخطوة الثانية، ثم اضغط على المفتاح Enter. سوف يتم فتح صفحتان ويب، ويوضح الشكل رقم (١٢,٦) أحد هذه الصفحات.



الشكل رقم (١٢,٦): صفحة ويب اختبار برنامج IIS.

٥- ولأنك تستخدم نفس جهاز الحاسب لكل من عميل الويب وخادم الويب، تستطيع أن تستخدم "localhost" بدلاً من اسم جهازك الحاسب (سوف نستخدم "localhost" في جميع أمثلة هذا الكتاب)، ولذلك اكتب "http://localhost/localstart.asp" داخل صندوق نص العنوان واضغط على المفتاح Enter، وستظهر صفحة الويب نفسها الظاهرة في الشكل رقم (١٢,٦).

■ لاحظ أنه إذا لم يظهر برنامج متصفح الويب صفحة ويب شبيهة بالشكل رقم (١٢,٦) راجع موضوع "Moving Web Applications and Solving Web Problems" في نهاية هذا الفصل للحصول على حل لهذه المشكلة.

التعامل مع مجلدات الويب
Working with Web Directories

لكي يتم الوصول إلى صفحة ويب مخزنة على خادم الويب بنجاح، يجب علينا أن نفهم كيفية تخزين الملف على الخادم. عندما يصل طلب صفحة ويب إلى خادم الويب من عميل الويب، فيجب على خادم الويب أن يصل لهذا الملف داخل نظام التخزين الخاص به بناء على المعلومات المرسلة في عنوان الصفحة المطلوبة. يوجد نوعان من المجلدات داخل خادم الويب وهما: مجلد واقعي (Physical Directory)، ومجلد افتراضي (Virtual Directory).

تشبه المجلدات الواقعية المجلدات التي نتعامل معها يومياً، وذلك بإنشاء ملفات داخلها وتخزينها (مثل ملفات التقارير)، أي عندما ننشئ مجلداً باستخدام برنامج مستكشف الويندوز (Windows Explorer) فهذا المجلد يطلق عليه مجلد واقعي مثل المجلد "C:\Chap12" حيث يمثل المجلد "Chap12" مجلداً واقعياً.

وعلى الجانب الآخر، إن المجلدات الافتراضية ليس لها وجود على المجلد الرئيس للموقع بل هي أسماء مستعارة تفسر بواسطة خادم الويب إلى مجلدات واقعية مناظرة. توجد ميزتان للمجلدات الافتراضية عن المجلدات الواقعية: أولاً، تستخدم عناوين أقصر مما يجعلها أسهل في التعامل. ثانياً، استخدام المجلدات التخيلية يخفي التركيب الداخلي لمجلدات الموقع الواقعية مما يساعد على تجنب الأخطار الأمنية.

ولأننا نستخدم المجلدات الافتراضية في معظم أعمالنا، تعال نتعلم كيف ننشئ مجلداً افتراضياً:

١- استخدام برنامج مستكشف الويندوز (أو نافذة My Computer) لكي ننشئ مجلداً واقعياً على القرص الصلب الخاص بك وليكن "C:\Chap12\Exercises\Ex01".

٢- اضغط بالزر الأيمن على المجلد Ex01، ثم اختر الأمر Sharing (إذا كنت تستخدم ويندوز XP فاختر "Sharing and Security")، فعندئذ ستظهر نافذة "Ex01 Properties".

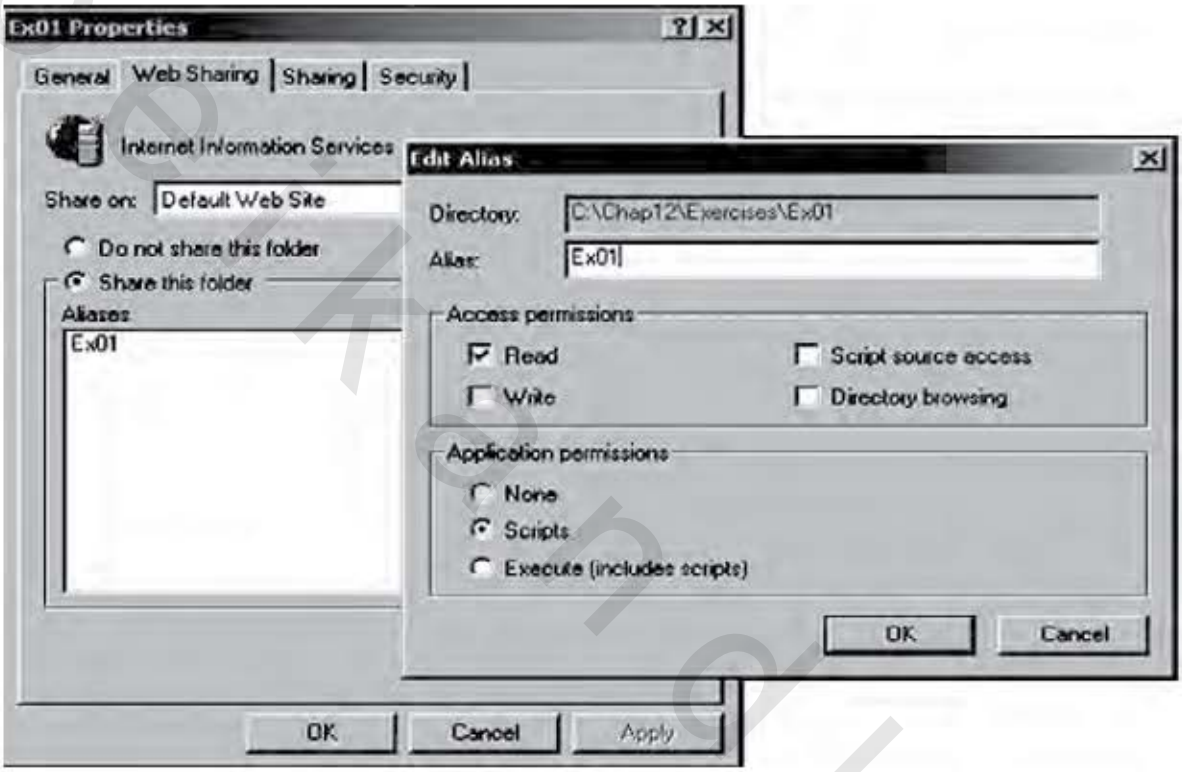
٣- اختر مسطرة "Web Sharing" ثم اختر "Share this folder"، فعندئذ ستفتح نافذة "Edit Alias" الموضحة في الشكل رقم (١٢,٧). استخدم تلك النافذة لإنشاء اسم مستعار (مجلد تخيلي) للمجلد الواقعي الحالي. وبشكل افتراضي يقترح المعالج اسم الفهرس الحالي نفسه ليستخدم كاسم مستعار (Ex01).

٤- اضغط على الزر OK مرتين لكي تكمل عملية تنفيذ أمر إنشاء المجلد الافتراضي "Ex01" للمجلد الواقعي "C:\chap12\Exercises\Ex01".

٥- وأخيراً أغلق كل النوافذ المفتوحة.

يمكنك الآن نسخ ملفات الموقع الخاص بك داخل المجلد الواقعي C:\Chap12\Exercises\Ex01، ثم تشير إليها عند استخدام عميل ويب بالمجلد الافتراضي. فمثلاً يمكن استدعاء الملف index.asp هكذا

"http:\\localhost\ex01\index.asp" مباشرة، ولاحظ أن العنوان لا يوضح التركيب الداخلي لشجرة مجلدات الموقع وأقصر طولاً من العنوان الواقعي.



الشكل رقم (١٢,٧). إنشاء فهرس افتراضي.

مراجعة مفاهيم لغة HTML
Reviewing HTML Basics

تتكون لغة html من علامات تمييز (Tags) التي تحدد شكل عناصر صفحة الويب، فعلى سبيل المثال توجد علامات تميز تحديد كيفية ظهور النص أسود عريض (Bold) أو مائل (Italic) أو عنوان (Heading) أو جزء من فقرة (Paragraph) وهكذا. يتم تعريف علامات تمييز لغة html مسبقاً والمسؤولة عن تحديد شكل عناصر صفحة الويب من نصوص وجداول وصور وغير ذلك، حيث تبدأ كل علامة تمييز بعلامة أصغر من (<) وتنتهي بعلامة أكبر من (>)، كما توجد العديد من علامات التمييز التي تتكون من زوجين من هذه العلامات (علامة البداية وعلامة النهاية). تستخدم علامة النهاية الشرطة المائلة للأمام (/) بين علامتي أصغر من (<) وأكبر من (>)، كما توجد العديد من علامات التمييز التي تحتوي على صفات (Attributes)، فعلى سبيل المثال يوضح الشكل رقم (١٢,٨) علامتي تمييز اللتان تستخدمان لإظهار نص ما كعنوان.

الشكل رقم (١٢,٨). شكل عنصر HTML.

يطلق على محتوى الشكل رقم (١٢,٨) بعنصر html (HTML Element)، ويشار إلى الجزء الذي يقع بين علامتي التمييز بالمحتويات (Content)، ولاحظ أن قيمة الصفة "ALIGN" (CENTER) سوف تؤثر على ظهور النص وتجعله يظهر في منتصف السطر، كما تحتوي لغة html على علامات تمييز فارغة (Empty Tags) أو ما تسمى بعلامات تمييز فردية (Single Tags) مثل العلامة (<BR>) التي تتسبب في إدراج سطر جديد داخل صفحة الويب. إن لغة html ليست لغة حساسة لحالة الأحرف (Not Case Sensitive) مع أننا نستخدم الأحرف الكبيرة (Capital) في النصف الأول من هذا الفصل لنفرق فقط علامات التمييز عن المحتويات.

اكتشاف وثيقة HTML

Exploring HTML Documents

تنتمي وثائق html إلى الملفات النصية (أو ملفات ASCII) والتي يمكن أن يتم إنشاؤها باستخدام أي برنامج محرر نصوص بسيط مثل برنامج المفكرة (Notepad)، وبالرغم من وجود العديد من البرامج التي يمكن أن تخزن الملفات من النوع html إلا أن لغة html الناتجة تلقائياً يصعب قراءتها، وعلى العكس إن أدوات برمجة الويب المرئية مثل البرنامج MS FrontPage أو برنامج Macromedia Dreamweaver تدعك تنشئ صفحات ويب باستخدام أسلوب "ما تراه سوف تحصل عليه" (What you see is what you get). تدعم تقنية ASP.NET هذا الأسلوب نفسه عند تصميم صفحات الويب.

يوضح الشكل رقم (١٢,٩) نموذج مستند html، يتكون السطر الأول من تعليق يُستخدم لتوثيق تصميم الصفحة، كما يحتوي الشكل على أقل علامات تمييز مطلوبة للتوضيح نموذج صفحة ويب (علامة html وعلامة Head وعلامة Title وعلامة Body).

```
<--! This is a comment line in an HTML file-->
<HTML>
<HEAD>
<TITLE>This appears on the browser title bar</TITLE>
</HEAD>
        …other relevant header information goes here…
<BODY>
        …the main part of the Web page goes here…
</BODY>
</HTML>
```

الشكل رقم (١٢,٩). نموذج وثيقة HTML.

تستخدم علامات التمييز <HTML/> و <HTML> لتوضيح بداية المستند ونهايته، أما عنصر Head فيستخدم لتوثيق صفحة الويب، وتمثل محتويات عنصر Title عنوان الوثيقة الذي يظهر أعلى متصفح الويب عند زيارة هذه الصفحة، ويحتوي العنصر Body على المعلومات التي ستظهر داخل الصفحة أمام المستخدم. يقدم الجدول رقم (١٢,١) قائمة لعناصر html الشائعة وأسلوب استخدامها، وسوف يتم شرح عناصر نماذج الويب لاحقاً، وبذلك لم يرد لها ذكر داخل هذا الجدول.

الجدول رقم (١٢,١). علامات HTML الشائعة.

Tag	Meaning	Useful Attributes
Character level		
<I> </I>	Italic	
<B> </B>	Bold	
<BIG> </BIG>	Font size one size larger than the current font	
<CITE> </CITE>	Citation from published material (italic)	
<CODE> </CODE>	Programming code	
<EM> </EM>	Emphasis, usually bold	
<S> </S>	Alternate for strikethrough	
<SMALL> </SMALL>	Font size one size smaller than the current font	
Block level		
<A> </A>	Anchor tag used for links	HREF, SRC
<BLOCKQUOTE> </BLOCKQUOTE>	Indents and encloses a block of text in quotes; requires 	
 	Line break (carriage return)	ALIGN, FONT
<DIV> </DIV>	Section or division	ALIGN, FONT
<H1> </H1>… <H6> </H6>	Six headline font sizes, H1 being the largest	ALIGN, FONT
<HR>	Horizontal rule	ALIGN, NOSHADE, WIDTH, COLOR, SIZE
<OL> </OL>	Ordered list	TYPE, VALUE
<P> </P>	Logical paragraph	ALIGN
<PRE> </PRE>	Prevents browser formatting	
<STRIKE></STRIKE>	Strikethrough content	
<SUB>	Subscript	
<SUP>	Superscript	
<UL> </UL>	Unordered list	TYPE

تابع الجدول رقم (١٢،١).

Tag	Meaning	Useful Attributes
Tables		
<TABLE> </TABLE>	Table	ALIGN, BGCOLOR, BORDER, CELLSPACING, CELLPADDING, WIDTH
<TH> </TH>	Table head	ALIGN, ROWSPAN, COLSPAN
<TR> </TR>	Table row	ALIGN, ROWSPAN, COLSPAN, VALIGN
<TD> </TD>	Table data	ALIGN, ROWSPAN, COLSPAN, VALIGN

يهمل متصفح الويب المسافات والسطور الفارغة، كما ينسق النص جهة اليسار بشكل تلقائي، ومن ثم يمكن استخدام علامات تمييز html لتنسيق النصوص وإضافة مسافات فارغة داخل صفحة الويب حيث تستخدم العلامة
 لإدخال سطر فارغ والرمز " " لوضع مسافة واستخدام الصفة "ALIGN="value لتنسيق النص جهة اليسار أو اليمين أو الوسط طبقاً لقيمة Value (left, center, right).

لاحظ أنه لكي يعمل تطبيق الويب بشكل صحيح على برنامج IIS فيجب علينا ضبطه على الاختيار "IIS Application"، ولعمل ذلك راجع آخر موضوع في هذا الفصل لمعرفة التفاصيل.

عزيزي القارئ تعال لنصمم أول صفحة ويب لك (صفحة ثابتة المحتوى)، ولعمل ذلك اتبع التالي:

١- افتح برنامج محرر النصوص مثل برنامج المفكرة.

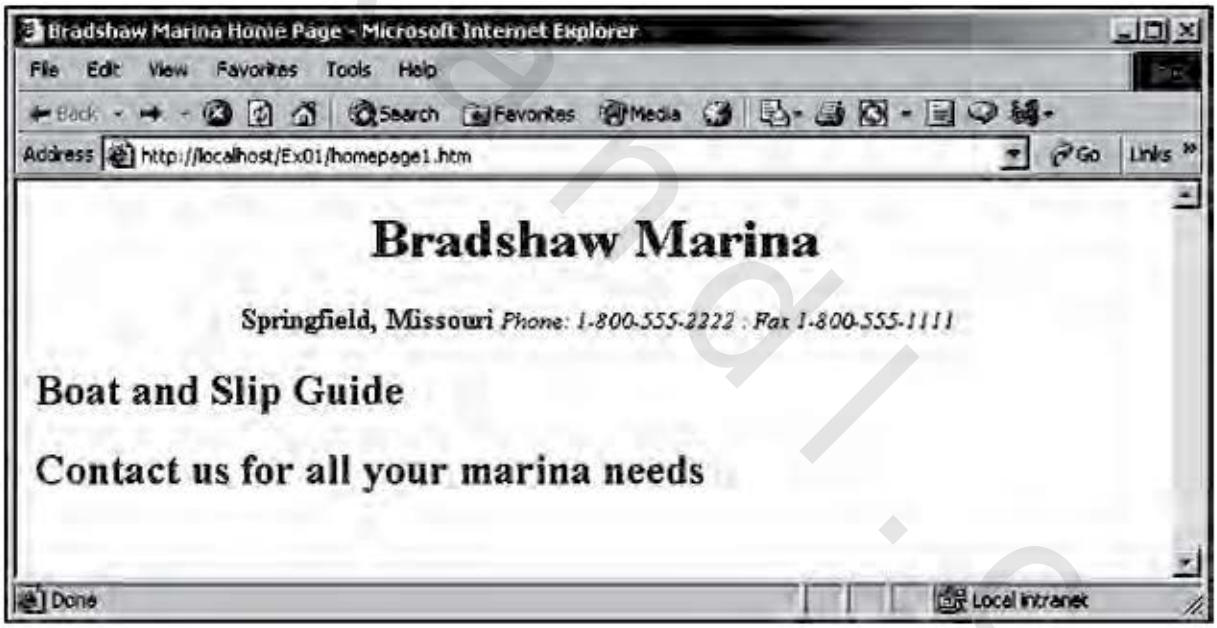
٢- اكتب النص التالي:

```
<HTML>
<HEAD>
<TITLE>  Bradshaw Marina Home Page</TITLE>
</HEAD>
<BODY>
<H1 ALIGN=CENTER>Bradshaw Marina</H1>
<P ALIGN=CENTER><B>Springfield, Missouri</B>
<SMALL><I>Phone: 1-800-555-2222    :
         Fax 1-800-555-1111</I></SMALL><BR>
<H2>Boat and Slip Guide</H2>
<H2>   Contact us for all your marina needs</H2>
</P>
</BODY>
</HTML>
```

٣- خزن هذا الملف بعنوان homepage1.htm داخل الفهرس Ex01 الذي أنشأناه سابقاً.

٤- ولعرض هذه الصفحة، افتح برنامج متصفح الإنترنت، ثم اكتب العنوان "http://localhost/Ex01/homepage1.htm"، ثم اضغط على المفتاح Enter.

يجب أن تتشابه صفحة الويب الظاهرة أمامك مع الشكل رقم (١٢,١٠) (لاحظ المسافات بين الكلمات والسطور وتنسيق النص داخل الصفحة) حيث يظهر كل من اسم المدينة واختصار اسم الولاية ورقم الهاتف ورقم الفاكس في نفس السطر مع أنها تظهر في أسطر متعددة داخل لغة html، كما أن المسافات الموجودة بينها داخل لغة html تم إهمالها عند ظهورها في صفحة الويب. سوف نعالج هذه المشكلة (عدم ظهور المسافات والسطور الفارغة لتنسيق النص) من خلال الخطوات القادمة.



الشكل رقم (١٢,١٠). صفحة ويب تحتوي على مسافات خاطئة.

عند تطوير صفحات الويب يفضل الاحتفاظ بكل من برنامج محرر النصوص وبرنامج متصفح الويب مفتوحين لكي تتمكن من رؤية التغيرات التي أجريتها على صفحة الويب بواسطة برنامج محرر النصوص داخل متصفح الويب وذلك بالضغط على زر "تحديث"، وإذا لم يظهر آخر تعديل في متصفح الويب فعليك بالضغط على زر "تخزين" في برنامج محرر النصوص (لعلك نسيت أن تخزن آخر تعديل)، ثم الضغط على زر "تحديث" في برنامج متصفح الويب. ولكي تتأكد أن ما يظهر في متصفح الويب هو صفحة الويب التي تقوم بتصميمها الآن، يمكنك رؤية شفرة المصدر باختيار قائمة View من برنامج متصفح الويب ثم اختيار الأمر Source.

والآن يمكنك تعديل صفحة homepage1.htm لكي تعالج مشكلة المسافات والسطور الفارغة:

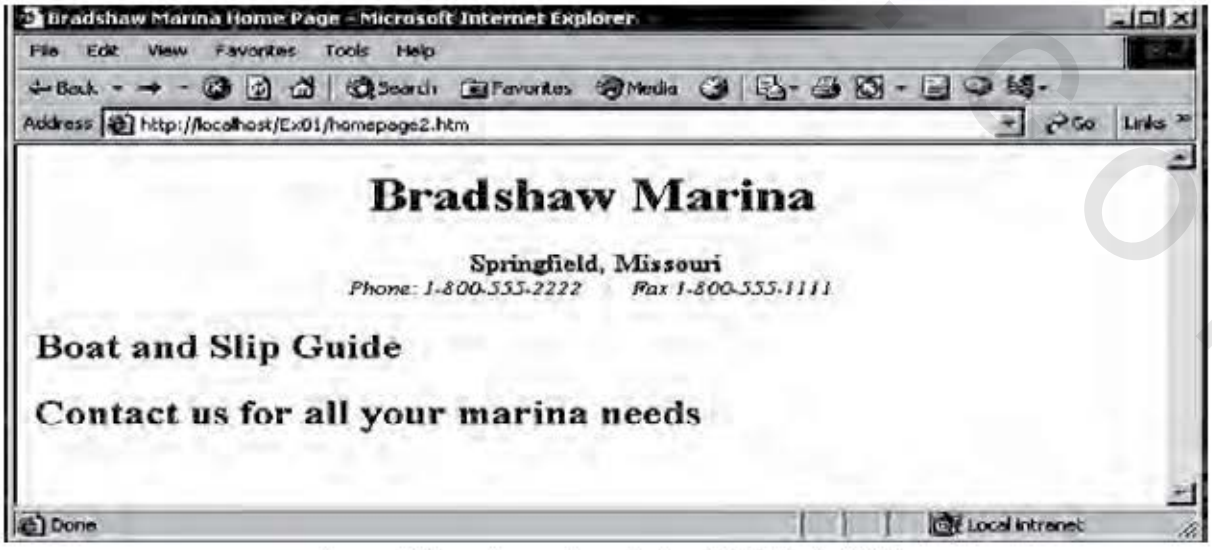
١- افتح الملف homepage1.htm داخل برنامج محرر النصوص.

٢- عدل المحتويات بإضافة شفرة html التي ستظهر بالخط الأسود العريض في النص التالي (لاحظ أن العلامة
 تعني سطراً فارغاً والرمز " " يعني إضافة مسافة):

```
<HTML>
<HEAD>
<TITLE>  Bradshaw Marina Home Page</TITLE>
</HEAD>
<BODY>
<H1 ALIGN=CENTER>Bradshaw Marina</H1>
<P ALIGN=CENTER><B>Springfield, Missouri</B>
<BR>
<SMALL><I>Phone: 1-800-555-2222     ;   
         Fax 1-800-555-1111</I></SMALL><BR>
<H2>Boat and Slip Guide</H2>
<H2>   Contact us for all your marina needs</H2>
</P>
</BODY>
</HTML>
```

٣- خزن الملف باسم جديد homepage2.htm وفي نفس المسار.

٤- اكتب العنوان "http://localhost/Ex01/homepage2.htm" داخل عنوان برنامج متصفح الويب، ثم اضغط على المفتاح Enter. يوضح الشكل رقم (١٢,١١) صفحة الويب التي يجب أن تظهر أمامك.



الشكل رقم (١٢,١١). صفحة ويب تحتوي على مسافات صحيحة.

التعامل مع الصور وروابط الاستدعاء
Working with Images and Hyperlinks

تستخدم علامة التمييز <IMG> لإضافة صور إلى صفحات الويب، وتستخدم الصفة SRC لكي تحتوي على موقع ملف الصورة، وتتنوع ملفات الصور التي يمكن إظهارها في صفحات الويب من النوع ".bmp" (Bitmap) والنوع ".gif" (Graphical Interchange Format) والنوع ".jpg" (Joint Photographic Express Group) وغيرها، مع العلم أن أكثر هذه الأنواع حجماً هو النوع ".bmp" ولذلك لا ينصح باستخدام هذا النوع مع صفحات الويب.

ولكي نوضح كيفية استخدام الصور داخل صفحات الويب، افترض أن لديك صورة مرسى مخزنة داخل الملف "sabtart_40.jpg". يجوز أن نكتب موقع الصورة داخل الصفة SRC بشكل نسبي (Relative) أو بشكل مطلق (Absolute) حيث ينسب الموقع النسبي إلى مسار المجلد الذي يحتوي على ملف الصورة إلى مجلد العمل الحالي، أما المسار المطلق فينسب إلى القرص الصلب. إذا وضعت ملف الصورة داخل المجلد Ex01، يستخدم سطر html التالي لإظهار الصورة في وسط الصفحة .

```
<IMG SRC="sabtart_40.jpg" ALIGN="CENTER">
```

أما إذا أنشأت مجلداً فرعياً داخل الفهرس Ex01 وأطلقت عليه اسم Images ثم وضعت داخله ملفات الصور، فيجب تغيير سطر html السابق ليصبح هكذا:

```
<IMG SRC="Images/sabtart_40.jpg" ALIGN="CENTER">
```

تستخدم رابطة الاستدعاء للانتقال إلى صفحة ويب أخرى داخل الموقع أو الانتقال إلى موقع آخر أو حتى الانتقال إلى برنامج البريد الإلكتروني، وتصاحب رابطة الاستدعاء نصاً أو صورة التي عندما يشير المستخدم إليها بزر الفأرة الأيسر يتغير شكله إلى يد كما يظهر النص باللون الأزرق ويظهر خط أسفله.

افترض أن صفحة الويب الحالية تحتوي في أسفل الصفحة على الموضوع الهام "All About Slips"، وتريد أن ينتقل إليه المستخدم مباشرة عند الضغط على رابطة استدعاء توجد في أعلى الصفحة. توضح الأوامر التالية كيف تنشئ رابطة لهذا الموضوع.

```
<A HREF="#Slips">For more slip information</A>
...
...
<A NAME="Slips">All About Slips</A>
```

تسمح علامة رابطة الاستدعاء <A> من خلال الصفة HREF أن تسند قيمة موقع محدد داخل صفحة الويب (مرجع) والتي ينتقل إليه عند الضغط على هذه الرابطة، ويستخدم الرمز # للإشارة إلى أن هذا المرجع يقع داخل حدود الصفحة الحالية، ولإنشاء هذا المرجع نستخدم الصفة Name التي يسند إليها قيمة هذا المرجع، فعلى سبيل المثال عند الضغط على رابطة الاستدعاء "For more slip information" يتم الانتقال مباشرة إلى الموضوع "All About Slips" الذي يقع داخل الصفحة نفسها.

وللانتقال إلى داخل صفحة ويب أخرى تقع داخل الموقع الخاص بك أو تقع في موقع آخر، نستخدم علامة التمييز HREF دون استخدام الرمز #، ويوضح سطر html التالي كيف نربط النص "Bradshaw Marina Boats" بصفحة الويب Sailboats.htm.

```
<A HREF="Sailboats.htm">Bradshaw Marina Boats</A>
```

تستطيع أيضاً أن تستخدم صورة (بدلاً من نص) كعنصر ربط، وذلك بوضع علامة تمييز الصورة <Img> داخل علامة تمييز <A> ، فعلى سبيل المثال يمكن ان نضع علامة التميز <Img> بدلاً من النص "Bradshow Marina Boats" في المثال السابق هكذا:

```
<A HREF="#Slips"><IMG SRC="Images/sabtart_40.jpg" ></A>
```

يمكنك الآن إضافة روابط استدعاء إلى الصفحة الرئيسة التي قمنا بإنشائها سابقاً، ولكن يجب عليك نسخ المجلد Images الذي يحتوي على الصور التي ستحتاجها خلال هذا الفصل من المجلد C:\Chap12\Examples من ملفات بيانات الكتاب إلى المجلد C:\Chap12\Exercises\Ex01، ولإضافة روابط الاستدعاء إلى الصفحة الرئيسة اتبع الآتي:

١- ابحث عن المجلد Ex01 داخل المجلد C:\Chap12\Examples الخاص بملفات الطلبة، ثم انسخ محتويات المجلد Ex01 إلى المجلد Chap12\Exercises\Ex01.

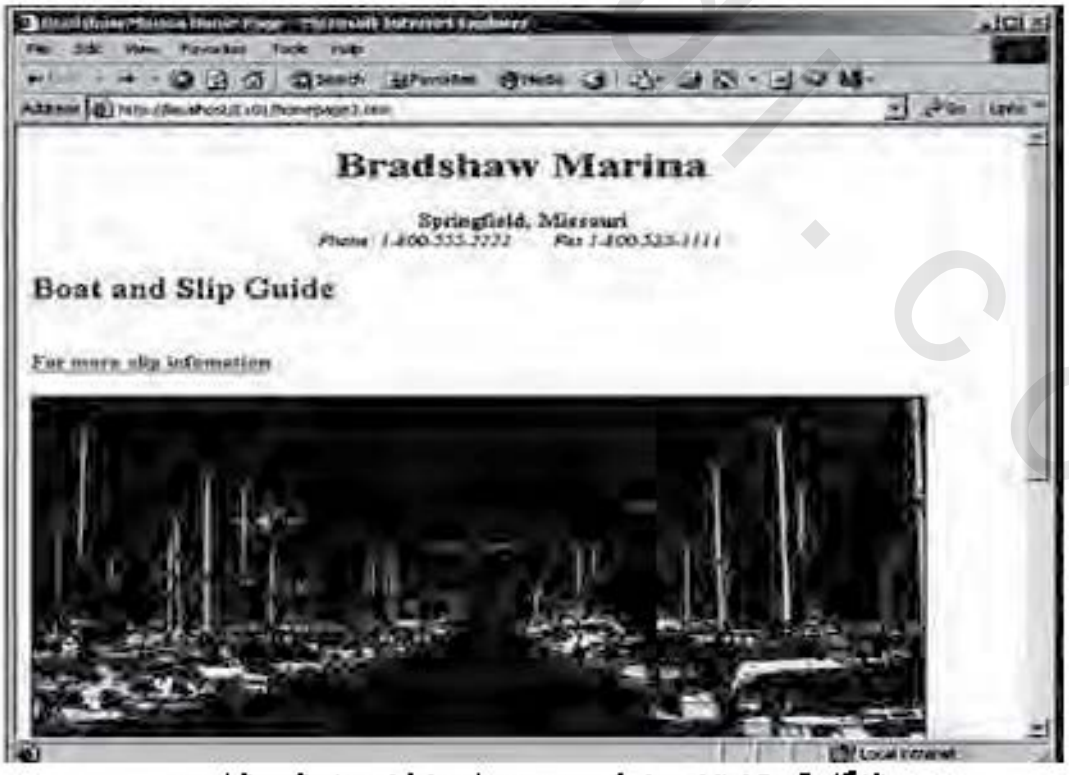
٢- افتح الملف homepage2.htm داخل برنامج المفكرة، ثم عدل محتوياته طبقاً للآتي:

```
<HTML>
<HEAD>
<TITLE>  Bradshaw Marina Home Page</TITLE>
</HEAD>
<BODY>
<H1 ALIGN=CENTER>Bradshaw Marina</H1>
<P ALIGN=CENTER><B>Springfield, Missouri</B><BR>
```

```
<SMALL><I>Phone: 1-800-555-2222     ;   
          Fax 1-800-555-1111</I></SMALL><BR>
<H2>Boat and Slip Guide</H2>
<BR>
<A HREF="#Slips"><B>For more slip information</B></A><BR><BR>
<A HREF="#Slips"><IMG SRC="images/subtart_40.jpg" ALT="Slip Info"></A><BR>

<H2>   Contact us for all your marina needs</H2>
<A HREF="Sailboats.htm">Sailboats</A><BR>
<A HREF="http://www.microsoft.com">Microsoft</A>
</P>
<BR>
<BR>
<BR>
<BR>
<H2><B><a NAME="Slips">all about Slips</A></B></H2>
</BODY>
</HTML>
```

٣- قم بتخزين الملف بالاسم الجديد homepage3.htm في نفس موقعه الحالي.

٤- اكتب العنوان "http://localhost/Ex01/homepage3.htm" داخل صندوق نص عنوان متصفح الويب، ثم اضغط على المفتاح Enter، فمنتئذ ستظهر صفحة الويب الموضحة في الشكل رقم (١٢,١٢).



الشكل رقم (١٢,١٢). صفحة ويب تحتوي على روابط استدعاء (النمرة العلوي).

٥- تتسبب رابطة الاستدعاء الأولى في هذه الصفحة إلى الانتقال إلى الموضوع "All About Slips" الذي يقع أسفل الصفحة نفسها (انظر الشكل رقم ١٢,١٣).

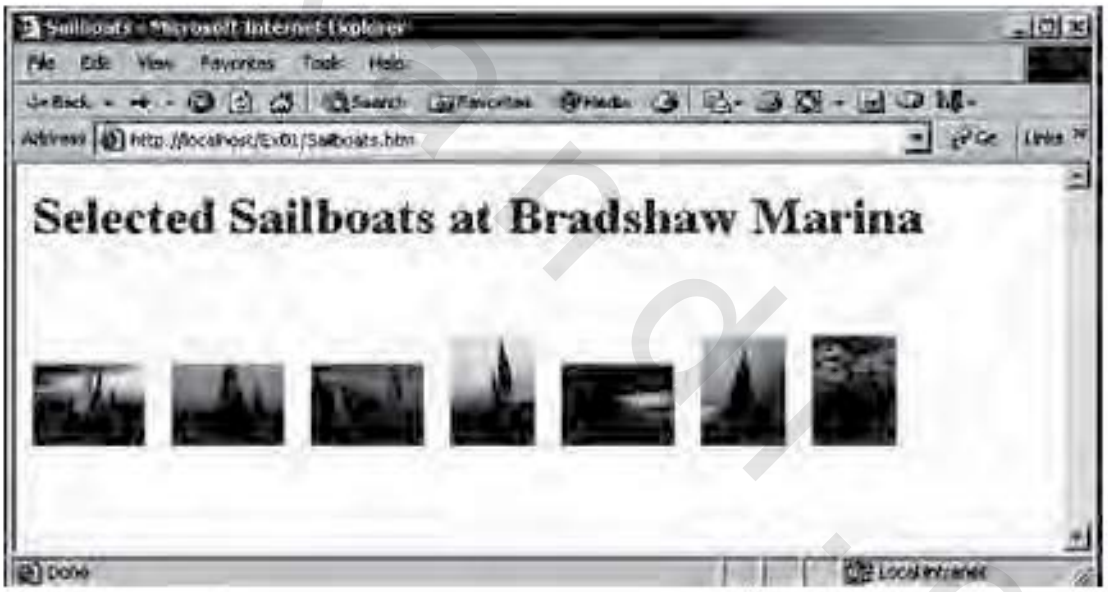
٦- تتسبب رابطة استدعاء الثانية (التي تستخدم صورة) إلى الانتقال إلى المكان نفسه ("All about slips#")، ولاحظ استخدام الصفة ALT التي أُلحقت بالعلامة <IMG> إلى إضافة معلومات حول الصورة (ToolTip) والتي تظهر عندما يشاور المستخدم على الصورة بالفأرة.

الشكل رقم (١٢,١٣). صفحة ويب تحتوي على روابط استدعاء (الجزء السفلي).

٧- استخدم أعمدة التصفح للوصول إلى قمة الصفحة ثانياً ثم شاور على الصورة، فعندئذ ستظهر معلومات الصفة ALT، ثم اضغط على الصورة لتنتقل مرة ثانية إلى الموضوع "All About Slips".

أما رابطة الاستدعاء الثالثة فستنقل بك إلى صفحة ويب جديدة تسمى "Sailboats.htm" التي تحتوي على صور المراكب كما هو موضح في الشكل رقم (١٢,١٤) حيث يجب أن يتم إنشاء هذا الملف أولاً، ثم يخزن طبقاً لشفرة HTML التالية.

```
<HTML>
<HEAD>
<TITLE>Sailboats</TITLE>
</HEAD>
<BODY>
<H1>Selected Sailboats at Bradshaw Marina</H1><HR>
<P>
<IMG SRC="images/sabtart_01.gif">   
<IMG SRC="images/sabtart_04.gif">   
<IMG SRC="images/sabtart_06.gif">   
<IMG SRC="images/sabtart_07.gif">   
<IMG SRC="images/sabtart_10.gif">   
<IMG SRC="images/sabtart_32.gif">   
<IMG SRC="images/sabtart_32.gif">
</BODY>
</HTML>
```



الشكل رقم (١٢,١٤). صفحة الويب Sailboats.htm.

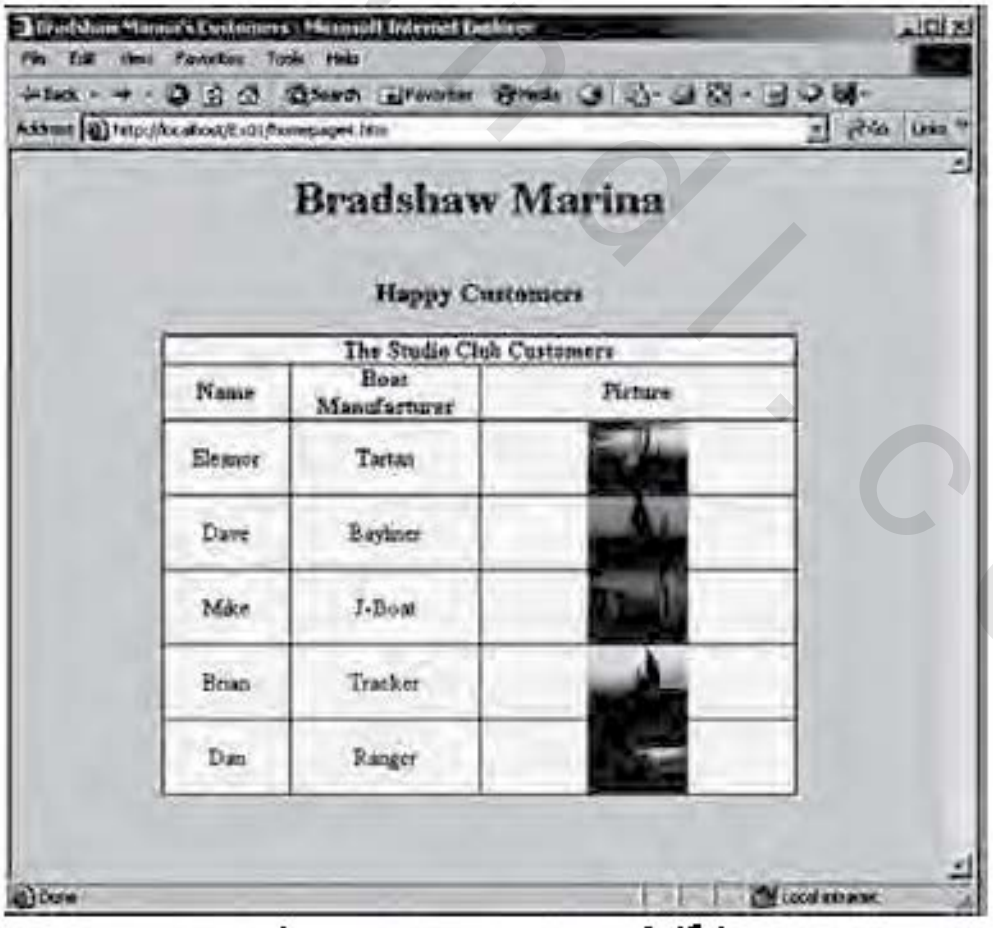
٨- ولكي تختبر صحة هذه الرابطة اضغط عليها، فعندئذ سينتقل بك متصفح الويب إلى الملف "Sailboats.htm"، ثم اضغط على الزر Back الموجود داخل صندوق أدوات متصفح الويب للعودة إلى الصفحة الرئيسة.

٩- أما رابطة الاستدعاء الأخيرة (Microsoft) فتنتقل بك إلى موقع شركة مايكروسوفت.

عزيزي القارئ يوجد لكل موقع ما يسمى بصفحة الويب الافتراضية (Default Page) التي لا يلزم كتابة اسم الملف الخاص بها عند زيارة الموقع، بل يكفي كتابة عنوان الموقع فقط، ولكن يلزم عليك أن تسمي ملف هذه الصفحة بالاسم default.htm أو بالاسم index.htm، مع العلم أن برنامج IIS يمكنك من تغيير اسم ملف الصفحة الافتراضية، ولكن جرى العرف أن تسمى بأحد هذين الاسمين، فعلى سبيل المثال يمكنك تغيير اسم الملف

homepage3.htm إلى الاسم default.htm، ومن ثم تستطيع أن تكتب عنوان هذه الصفحة هكذا "http://localhost/Ex01" بدلاً من كتابته كاملاً.

إن موضوع تنسيق المعلومات داخل صفحة الويب يمثل مشكلة لمصمم مواقع الإنترنت، ولحل هذه المشكلة يمكن أن تستخدم الجداول كأداة تنسيق أو لعرض معلومات جدولية. تشبه جداول صفحات الويب جداول برنامج MS Word حيث يتكون الجدول من صف العنوان وصفوف البيانات، كما يمكن دمج أعمدة سوية أو دمج صفوف سوية. يمكن إدراج نص أو صور أو روابط استدعاء داخل الجداول، وإذا أردت أن تظهر الجدول دون إظهار حدوده فيمكنك القيام بذلك من خلال إسناد القيمة صفر إلى الصفة Border داخل علامة تمييز الجدول <TABLE>، ولكي تستوعب شفرة HTML الخاصة بالجداول انظر إلى الشكل رقم (١٢,١٥) الذي يظهر بيانات العملاء في الجدول، كما يوضح الشكل رقم (١٢,١٦) شفرة HTML المناظرة لهذه الصفحة (ملف هذه الصفحة هو homepage4.htm).



الشكل رقم (١٢,١٥). صفحة ويب تستخدم جدول.

```
<HTML>

<HEAD>
<TITLE>Bradshaw Marina's Customers</TITLE>
</HEAD>

<BODY BGCOLOR="$00FFFF">
<H1 ALIGN=CENTER>Bradshaw Marina</H1>
<BR>
<TABLE BGCOLOR="yellow" BORDER="1" CELLPADDING="0" CELLSPACING="0"
       STYLE="border-collapse: collapse" BORDERCOLOR="$111111"
       WIDTH ="70%" ALIGN=CENTER ID="BradshawMarina1"
       ALIGN=CENTER>
<CAPTION ALIGN=TOP><H3>Happy Customers</H3></CAPTION>
<TH COLSPAN=3>The Studio Club Customers</TH>
<TR ALIGN=CENTER>
 <TH>Name</TH>
 <TH>Boat Manufacturer</TH>
 <TH>Picture</TH>
</TR>
<TR ALIGN=CENTER>
 <TD WIDTH="20%">Eleanor</TD>
 <TD WIDTH="30%">Tartan</TD>
 <TD WIDTH="50%" VALIGN=TOP>
    <IMG SRC="Images/sabtart_01.GIF" WIDTH="68" HEIGHT="51"></TD>
</TR>
<TR ALIGN=CENTER>
 <TD WIDTH="20%">Dave</TD>
 <TD WIDTH="30%">Bayliner</TD>
 <TD WIDTH="50%" VALIGN=TOP>
       <IMG SRC="Images/sabtart_04.GIF" WIDTH="68" HEIGHT="51"></TD>
</TR>
<TR ALIGN=CENTER>
 <TD WIDTH="20%">Mike</TD>
 <TD WIDTH="30%">J-Boat</TD>
 <TD WIDTH="50%" VALIGN=TOP>
       <IMG SRC="Images/sabtart_06.GIF" WIDTH="68" HEIGHT="51"></TD>
</TR>
<TR ALIGN=CENTER>
 <TD WIDTH="20%">Brian</TD>
 <TD WIDTH="30%">Tracker</TD>
 <TD WIDTH="50%" VALIGN=TOP>
       <IMG SRC="Images/sabtart_07.GIF" WIDTH="68" HEIGHT="51"></TD>
</TR>
<TR ALIGN=CENTER>
 <TD WIDTH="20%">Dan</TD>
 <TD WIDTH="30%">Ranger</TD>
 <TD WIDTH="50%" VALIGN=TOP>
       <IMG SRC="Images/sabtart_19.GIF" WIDTH="68" HEIGHT="51"></TD>
</TR>
</TABLE>

</BODY>

</HTML>
```

الشكل رقم (١٢,١٦)- شفرة صفحة ويب المعتمدة على جدول.

obeikandl.com

لاحظ أن لون خلفية صفحة الويب تم تحديدها داخل علامة التمييز <BODY> حيث تستخدم صيغة الأرقام السداسية عشر (كما فعلنا في لون خلفية الصفحة) أو أسماء الألوان (متغيرات ثابتة) لإسناد قيمة الألوان كما سنفعل لإسناد لون خلفية الجدول، وتوضح شفرة HTML التالية تعريف الجدول باستخدام علامة التمييز <TABLE>.

```
<TABLE BGCOLOR="yellow" BORDER="1" CELLPADDING="0" CELLSPACING="0"
      STYLE="border-collapse: collapse" BORDERCOLOR="#111111"
      WIDTH ="70%" ALIGN=CENTER ID="BradshawMarina1"
      ALIGN=CENTER>
```

تستخدم الصفة BORDER لإسناد قيمة سمك حدود الجدول، مع العلم أنه إذا أسندنا القيمة صفر لهذه الصفة سيتم إلغاء الحدود، وتستخدم الصفة CELLSPACING في ضبط المسافة التي تقع بين حدود خلايا الجدول، وتستخدم الصفة CELLPADDING في ضبط المسافة التي تقع بين حد الخلية الداخلية ومحتويات الخلية في الجدول، وتستخدم الصفة BORDERCOLOR لإسناد لون حدود الجدول الخارجية، كما تستخدم الصفة WIDTH لإسناد قيمة عرض الجدول التي ربما تأخذ نسبة مئوية من عرض الصفحة التي سيظهر داخلها الجدول أو تأخذ قيمة مطلقة. لاحظ أيضاً أن الجدول لا يظهر في وسط الصفحة، كما أن الجدول يحتوي على الصفة ID التي تحدد اسم الجدول والذي يستخدم كمرجع للجدول لاحقاً.

توضح الأوامر التالية شفرة HTML المسؤولة عن تحديد عنوان (Caption) للجدول وعناوين أعمدة الجدول.

```
<CAPTION ALIGN=TOP><H3>Happy Customers</H3></CAPTION>
<TH COLSPAN=3>The Studio Club Customers</TH>
<TR ALIGN=CENTER>
   <TH>Name</TH>
   <TH>Boat Manufacturer</TH>
   <TH>Picture</TH>
  </TR>
```

يمكن إظهار عنوان الجدول أسفل الجدول وذلك بإسناد القيمة BOTTOM للصفة ALIGN، وتشير علامة التمييز <TH> </TH> إلى رأس الجدول حيث تم دمج أعمدة رأس الجدول الثلاثة (Colspan = 3)، وتشير علامة التمييز <TR> </TR> إلى إدراج صف جديد في الجدول، وتشير علامة التمييز <TD> </TD> إلى خلايا الجدول حيث يتم وضع البيانات.

وعلى هذا النحو يتم تصميم الجدول حيث توجد خمسة صفوف تحتوي على بيانات خمسة عملاء، وتوضح أوامر شفرة HTML التالية بيانات العميل الأول.

```
<TR ALIGN=CENTER>
  <TD WIDTH="20%">Eleanor</TD>
  <TD WIDTH="30%">Tartan</TD>
  <TD WIDTH="50%" VALIGN=TOP>
     <IMG SRC="Images/sabtart_01.GIF" WIDTH="68" HEIGHT="51"></TD>
</TR>
```

تستخدم الصفة WIDTH لتحديد عرض كل عمود داخل الجدول، وتستخدم الصفة VALIGN لتنسيق الصورة رأسياً داخل خلية الجدول، كما تحتوي الصورة على كل من الصفة WIDTH لتحديد عرض الصورة والصفة LENGTH لتحديد طول الصورة.

سوف يحتوي هذا الفصل على أساسيات لغة HTML عدا موضوع الإطارات (Frames)، وهذا بسبب أن تقنية ASP.NET تحتوي على بديل جيد لهذا الموضوع الذي سيتم مناقشته في الفصل السادس عشر، كما أن موضوع "إدراج ملفات تعمل لدى الخادم" (Server-Side Includes) لم تيم عرضه أيضاً لأنه يوجد مفهوم بديل داخل تقنية ASP.NET يسمى "Code-Behind" والذي سنشرحه لاحقاً في هذا الفصل. أما موضوع توحيد مظهر صفحات الويب الذي يهتم بتعريف ملف CSS والذي يحتوي على جميع مواصفات مظهر صفحات الويب فهو خارج نطاق هذا الكتاب.

استخدام نماذج HTML
Using HTML Forms

تستخدم نماذج HTML لتجميع بيانات من المستخدم لإرسالها إلى خادم الويب ومن ثم معالجتها، ثم إرسال النتيجة إلى العميل، ويمكنك وضع أكثر من نموذج (علامة التمييز <form> </form>) في وثيقة HTML واحدة.

يحتوي عنصر نموذج HTML على ثلاث صفات. أولاً: الصفة NAME التي تستخدم لإسناد اسم النموذج والذي يستخدم كمرجع للنموذج، ومن ثم نستطيع أن نتعامل مع عناصر النموذج من خلال هذا الاسم (لأنه كما ذكرنا سابقاً ربما تحتوي صفحة الويب على أكثر من نموذج). ثانياً: الصفة METHOD التي تأخذ القيمة "POST" أو القيمة "GET" حيث تسند القيمة "POST" إذا أردت أن ترسل بيانات النموذج إلى خادم الويب داخل مجموعة منفصلة تسمى Form والتي تنتمي إلى الكائن Request، أما إذا استخدمت القيمة "GET" فسوف ترسل بيانات النموذج إلى خادم الويب بإضافتها إلى نهاية عنوان صفحة الويب المستدعاة، وبذلك ستظهر البيانات أمام المستخدم (تخزن البيانات في هذه الحالة داخل مجموعة تسمى QueryString والتي تنتمي أيضاً إلى الكائن Request). تأخذ

بيانات النموذج المرسلة (سواء باستخدام أسلوب POST أو أسلوب GET) الصيغة "NAME="Value المعروفة حيث يمثل NAME أحد عناصر النموذج وتمثل Value البيانات التي أدخلها المستخدم داخل هذا العنصر، فعلى سبيل المثال إن العبارة fName=Eleanor تعني أن صندوق النص fName يحتوي على القيمة Eleanor. ثالثاً: الصفة ACTION التي تخزن اسم ملف البرنامج الذي سيستقبل بيانات النموذج والذي سينفذ لدى الخادم، مع العلم أن العادة جرت استخدام ملف آخر غير ملف النموذج، ولكن لا مانع من استخدام ملف النموذج نفسه.

استخدام عناصر النموذج
Using Form Controls

تتشابه عناصر واجهة استخدام نماذج الويب مع عناصر واجهة استخدام البرامج التي تعمل تحت بيئة الويندوز بشكل عام، ويلخص الجدول رقم (١٢,٢) عناصر نماذج ويب.

إن معظم عناصر نماذج HTML يوجد لها نظير في تقنية ASP.NET والتي تتشابه مع عناصر لغة VB .NET التي استخدمناها سابقاً لتطوير تطبيقات الويندوز. يستجيب عنصر الزر (Button) داخل لغة HTML إلى الحدث ONCLICK والذي يحدث عند ضغط المستخدم عليه بزر الفأرة الأيسر. فعلى سبيل المثال، نرى أن العنصر الزر (الموضح في الجدول رقم ١٢,٢) يستجيب للحدث ONCLICK باستدعاء الإجراء "btnCheck"، مع العلم أنه يمكنك كتابة الأوامر مباشرة بين علامتي التنصيص داخل علامة التمييز <INPUT> أو كتابة اسم إجراء بين علامتي التنصيص (مثل الإجراء "btnCheck").

الجدول رقم (١٢,٢). محتويات عناصر نماذج HTML.

INPUT Type or Tag and Typical Attributes	VB .NET Equivalent
<INPUT TYPE= "Text" NAME= "txtName" SIZE = "15" >	Text box
<INPUT TYPE= "Password" NAME ="txtPass" SIZE = "12" >	Password
<INPUT TYPE= "Radio" NAME = "rdbPay" VALUE="MONTHLY"> <INPUT TYPE= "Radio" NAME = "rdbPay" VALUE= "SEMI" > <INPUT TYPE= "Radio" NAME = "rdbPay" VALUE= "ANNUALLY" CHECKED>	Radio buttons
<INPUT TYPE = "Checkbox" NAME="chkSBE" VALUE="sb" CHECKED > <INPUT TYPE = "Checkbox" NAME="chkSKBE" VALUE="skb"> <INPUT TYPE = "Checkbox" NAME="chkCCE" VALUE="cc">	Check boxes
<INPUT TYPE = "Button" NAME ="btnCheck" ONCLICK="btnCheck" VALUE="OK">	Button
<INPUT TYPE = "Submit">	None
<INPUT TYPE = "Reset">	None
<INPUT TYPE= "Hidden" NAME= "txtMessage" VALUE= "True" >	None
<TEXTAREA ROWS="2" NAME="txtWant" > ... ... </TEXTAREA>	Multiline text box
<SELECT NAME="cboColor" <OPTION>"Red"</OPTION> <OPTION>"Blue"</OPTION> <OPTION>"Green"</OPTION> </SELECT>	Combo box

لا يوجد لكل من الزر SUBMIT والزر RESET (عناصر لغة HTML) نظير في تقنية ASP.NET. يأخذ الزر SUBMIT العنوان "Submit" ويستخدم لإرسال بيانات النموذج إلى خادم الويب، ويأخذ الزر RESET العنوان "Reset" ويستخدم لمسح عناصر النموذج، كما يستخدم العنصر HIDDEN (صندوق نص مخفي) لتبادل المعلومات بين صفحات الموقع. يسرد الجدول رقم (١٢,٣) أهم أحداث لغة HTML الشائعة بالإضافة إلى الحدث ONCLICK.

الجدول رقم (١٢,٣). أحداث HTML.

HTML Event	Description
OnBlur	Control loses focus
OnChange	Control loses focus and the value has changed
OnClick	Control has been clicked
OndbClick	Control has been double-clicked
OnFocus	Control receives focus
OnKeyDown	Key has been pressed with the control having focus
OnKeyPress	Key has been pressed and released with the control having focus
OnKeyUp	Key has been released with the control having focus
OnMouseDown	Left mouse key pressed while over the control
OnMouseMove	Mouse moves while over the control
OnMouseOut	Mouse moves away from the control
OnMouseOver	Mouse moves over the control
OnMouseUp	Left mouse key released while the control has focus
OnReset	Resets the form removing all user inputs (Reset button)
OnSubmit	Submits the form (Submit button)

تطوير تطبيقات الويب باستخدام تقنية ASP

Classic ASP Web Development

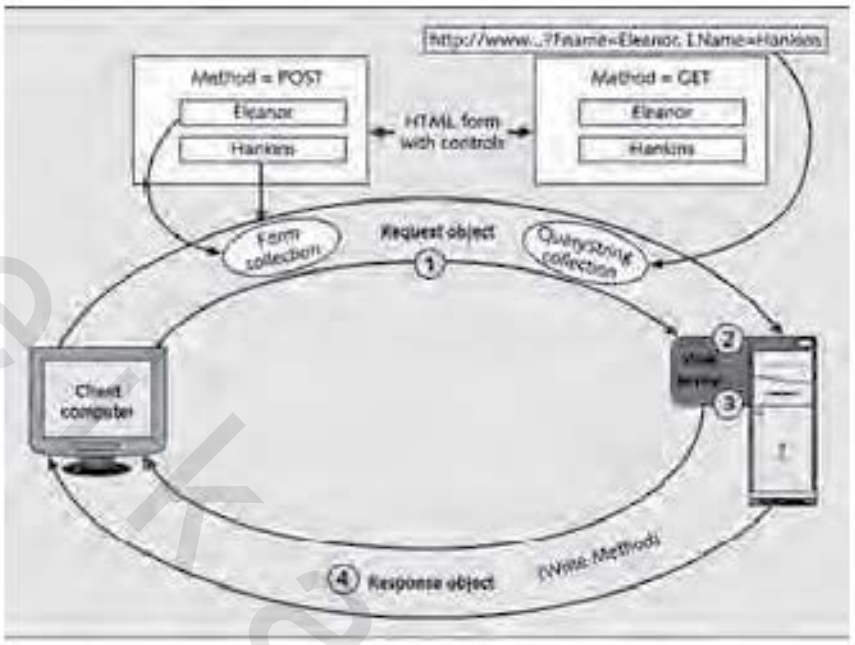
ظلت تقنية ASP لسنوات عديدة أحد أهم التقنيات شيوعاً في تطوير تطبيقات الويب، وبعد تقديم إطار عمل نت وتقنية ASP.NET كجزء من فيجوال ستوديو نت تغير اسم تقنية ASP إلى "Classic ASP"، ومع أن تقنية ASP.NET تم تغييرها وتحسين أسلوب تطوير تطبيقات الويب إلا أنها مازالت تعتمد على تقنية ASP وتستخدم أجزاء منها؛ ولذلك فإن مراجعة تقنية ASP يساعدنا على فهم تطبيقات الويب بشكل عام وتقنية ASP.NET بشكل خاص، ومازال كل من إطار عمل نت وفيجوال ستوديو نت تدعم صيانة تطبيقات الويب وتطويرها باستخدام تقنية "Classic ASP".

تسمح تقنية ASP أن تدخل أوامر نصية (Scripts) داخل شفرة HTML والتي سوف تنفذ لدى خادم الويب باستخدام محرك تنفيذ أوامر ASP (Server-Side)، ولكن يجب عليك أن تغير الاسم الممتد للملف من htm. إلى asp.، وإن لم تقم بتغيير الاسم الممتد فلن يتم تنفيذ أوامر ASP الموجودة داخل ملف ASP، كما يجب أن تلاحظ أن ملفات HTML ذات الامتداد htm. يمكن أن تحتوي على أوامر نصية (Scripts) تنفذ لدى العميل (Client-Side)، وفي هذه الحالة لا يتطلب إعادة تسميتها لتكون ذات امتداد asp..

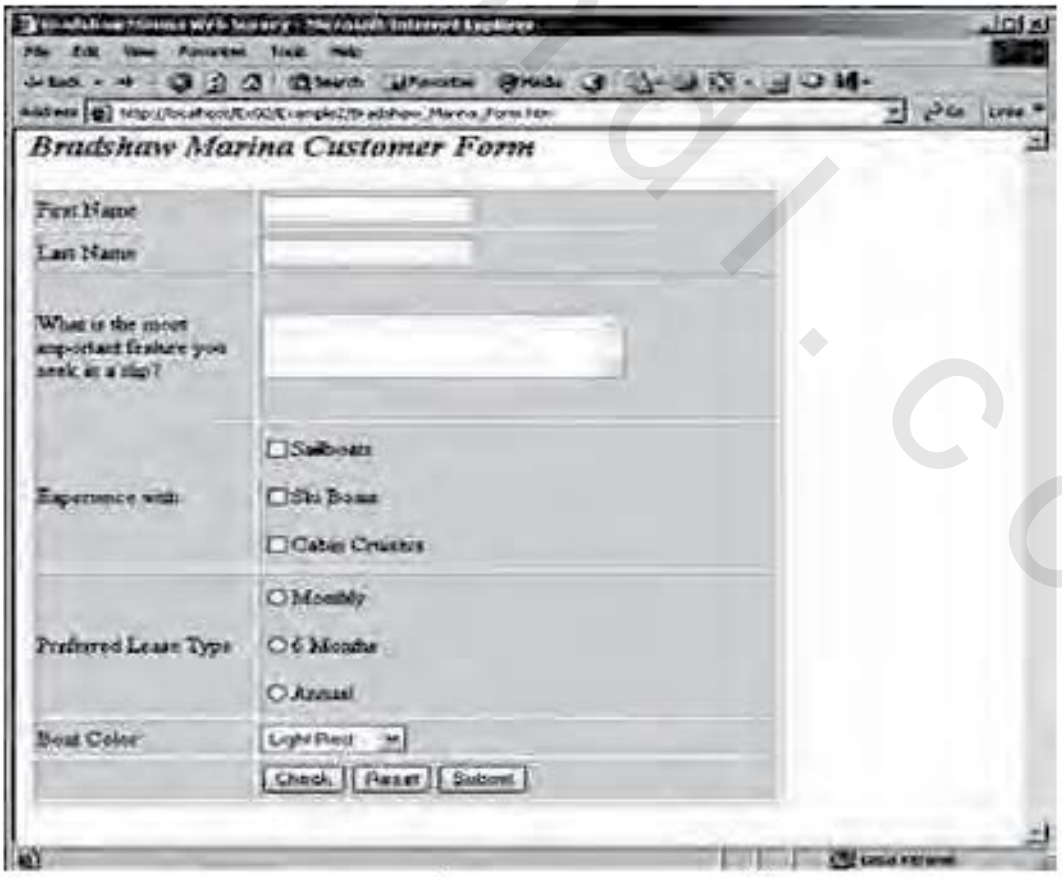
تسمح تقنية ASP بكل من المعالجة لدى العميل (Client-Side Processing) والمعالجة لدى الخادم (Server-Side Processing) باستخدام عدة لغات نصية (Scripting Languages) مثل لغة VBScript ولغة JavaScript، وعادة تستخدم المعالجة لدى العميل للتحقق من صحة البيانات قبل إرسالها إلى الخادم كما يمكن أن تستخدم لعمل تأثيرات في مظهر عناصر صفحة الويب بعد تحميلها مثل حركة مجموعة صور، بينما تستخدم البرمجة لدى الخادم لاستقبال بيانات من المستخدم ثم استرجاع بيانات من قاعدة البيانات (إذا كانت هناك حاجة لذلك) ثم المعالجة وعودة إرسال نتيجة المعالجة إلى عميل الويب في صيغة لغة HTML، ويوضح الشكل رقم (١٢.١٧) نموذج Request/Response المستخدم للتعامل بين كل من عميل الويب وخادم الويب. لاحظ أن المستخدم يدخل أولاً البيانات في نموذج، ثم يضغط على الزر SUBMIT التي يرسل البيانات المدخلة في مجموعة Form (إذا كانت METHOD="Post") أو مجموعة QueryString (إذا كانت METHOD="Get")، ثم يقوم خادم الويب بالمعالجة ثم يرسل نتيجة المعالجة إلى عميل الويب باستخدام الإجراء Write الذي ينتمي إلى الكائن Response.

توجد كائنات أخرى غير كل من الكائن Request والكائن Response داخل مكتبة ASP تستخدم بواسطة خادم الويب مثل الكائن Session والكائن Application الذين يستخدمان لمتابعة حالة كل مستخدم، ومن ثم تطبيق مفهوم برامج الويب لأن برتوكول HTTP لا يهتم بمتابعة حالة المستخدم. سوف نركز الآن على كل من الكائن Request والكائن Response، وسوف نتعرض للكائنات الأخرى بالتفصيل في الفصل السادس عشر.

افترض معي أن إدارة شركة برادشو مارينا أرادت أن تنشر نموذج ويب على موقعها يستخدم لاستقبال معلومات عن العملاء الحاليين وعن مراكبهم (الشكل رقم ١٢.١٨)، وعند الضغط على الزر SUBMIT سوف ترسل بيانات هذا النموذج إلى صفحة أخرى للمعالجة.



الشكل رقم (١٢,١٧). نموذج RequestResponse.



الشكل رقم (١٢,١٨). صفحة ويب شركة برادشو مارينا.

ولأن نموذج الويب السابق يحتوي على شفرة HTML، إذن يمكن أن يحتوي على أوامر نصية تعمل لدى العميل، وكما نعلم أن النموذج يمكن أن يظهر في أي مكان داخل صفحة الويب، مع العلم أن هذا النموذج يظهر أسفل الصفحة، ويوضح الشكل رقم (١٢,١٩) شفرة HTML المطلوبة لإنشاء هذا النموذج حيث تحتوي علامة التمييز <form> على الصفة ACTION التي تحدد ملف ASP الذي سوف ينفذ لدى الخادم عند إرسال هذا النموذج، كما تحتوي على الصفة METHOD التي تقوم بإرسال محتويات النموذج داخل المجموعة Form، وأخيراً تحتوي على الصفة NAME لتخزين اسم النموذج، أما بقية شفرة HTML تحتوي على شفرة عناصر النموذج HTML وشفرة الجدول التي يستخدم كأداة تنسيق عناصر النموذج.

```
<form name=frmTest action=Bradshaw_Marina_response.asp method=post>
<table border="1" width="72%" height="505">
  <tr>
    <td width="25%" bgcolor="#FFCC66" height="23">First Name</td>
    <td width="70%" bgcolor="#FFCC66" height="23">
        <input type="text" name="txtFirstName" size="20" tabindex="0" maxlengtn="20"></td>
  </tr>
  <tr>
    <td width="25%" bgcolor="#FFCC66" height="23">Last Name</td>
    <td width="70%" bgcolor="#FFCC66" height="23">
         <input type="text" name="txtLastName" size="20" tabindex="1" maxlength="20"></td>
  </tr>
  <tr>
    <td width="25%" bgcolor="#FFCC66" height="104">What is the most important
      feature you seek in a slip?</td>
    <td width="70%" bgcolor="#FFCC66" height="104"><textarea rows="3"
         name="txtDesire" cols="28" tabindex="2"></textarea></td>
  </tr>
  <tr>
    <td width="25%" bgcolor="#FFCC66" height="95">Experience with </td>
    <td width="70%" bgcolor="#FFCC66" height="95">
    <input type="checkbox" name="chkSailBoat" value="sb" tabindex="3">Sail
    Boats
      <p><input type="checkbox" name="chkSkiBoat" value="skb" tabindex="4">Ski
      Boats
      </p>
      <p><input type="checkbox" name="chkCabinCruiser" value="cc" tabindex="5">Cabin
      Cruisers
      </p>
    </td>
  </tr>
  <tr>
    <td width="25%" bgcolor="#FFCC66" height="95">Preferred Lease Type</td>
    <td width="70%" bgcolor="#FFCC66" height="95">
         <input type="radio" name="rdbType" id=radio1 value="monthly" tabindex="6">Monthly
     <p><input type="radio" name="rdbType" id=radio2 value="semi" tabindex="7">6 Months </p>
     <p><input type="radio" name="rdbType" id=radio3 value="annually" tabindex="8">Annual</p>
    </td>
  </tr>
  <tr>
    <td width="30%" bgcolor="#FFCC66" height="23">Boat Color</td>
    <td width="70%" bgcolor="#FFCC66" height="23">
    <select size="1" name="cboCheckColor" tabindex="9">
        <option value="Red">Light Red</option>
        <option value="Blue">Light Blue</option>
```

الشكل رقم (١٢,١٩). شفرة نموذج صفحة ويب شركة برادشو مارينا.

```
        <option value="Green">Light Green</option>
        <option value="Yellow">Light Yellow</option>
      </select></td>
  </tr>
  <tr>
    <td width="25%" bgcolor="FFFCC66" height="25"> </td>
    <td width="70%" bgcolor="FFFCC66" height="25">
    <input type="button" value="Check" name="btnCheck" tabindex="10" >
    <input type="reset" value="Reset" name="btnReset" tabindex="11">
    <input type="submit" value="Submit" name="btnSubmit" tabindex="11"></td>
  </tr>
</table>
<input type="Hidden" name="Message" value = "True">
</form>
</body>
</html>
```

تابع الشكل رقم (١٢,١٩).

ولتوضيح شفرة HTML الخاصة بعناصر النموذج فقد تم إزالة شفرة HTML الخاصة بصفوف الجدول كما هو واضح من الشكل رقم (١٢,٢٠). تحتوي جميع العناصر على الصفة NAME و الصفة TABINDEX، وقد أسندت مقدمة اسم مناسبة لكل على عنصر HTML، فعلى سبيل المثال أسندت المقدمة txt لصناديق النص وهكذا، وكما هو واضح من الشكل رقم (١٢,٢٠) يحتوي أول سطرين من شفرة HTML على شفرة صندوقي نص لاستقبال الاسم الأول والاسم الأخير للعميل، مع العلم أن حجم كل صندوق نص يساوي ٢٠ حرفاً فقط، أما عنصر نافذة النص TEXTAREA فهي تشبه عنصر صندوق نص متعدد الأسطر في لغة فيجوال بيسك نت. يحتوي هذا العنصر على الصفة ROWS التي تحدد عدد السطور، ثم تأتي صناديق التأشير حيث أسندنا قيمة مختلفة للصفة VALUE لكل صندوق تأشير. تذكر أن صناديق التأشير تستخدم عندما نريد أن نجعل المستخدم يختار خياراً أو أكثر من مجموعة خيارات.

```
<input type="text" name="txtFirstName" size="20" tabindex="0" maxlength="20">
<input type="text" name="txtLastName" size="20" tabindex="1"
maxlength="20">
<textarea rows="3" name="txtDesire" cols="28" tabindex="2">
<input type="checkbox" name="chkSailBoat" value="sb" tabindex="3">Sailboats
<input type="checkbox" name="chkSkiBoat" value="skb" tabindex="4">Ski
      Boats
<input type="checkbox" name="chkCabinCruiser" value="cc" tabindex="5">Cabin
Cruisers
<input type="radio" name="rdbType" id=radio1 value="monthly" tabindex="6">Monthly
     <p><input type="radio" name="rdbType" id=radio2 value="semi"
         tabindex="7">6 Months </p>
     <p><input type="radio" name="rdbType" id=radio3 value="annually"
         tabindex="8">Annual
<select size="1" name="cboCheckColor" tabindex="9">
        <option value="Red">Light Red</option>
        <option value="Blue">Light Blue</option>
        <option value="Green">Light Green</option>
        <option value="Yellow">Light Yellow</option>
</select>
<input type="button" value="Check" name="btnCheck" tabindex="10" >
<input type="reset" value="Reset" name="btnReset" tabindex="11">
<input type="submit" value="Submit" name="btnSubmit" tabindex="12">
<input type="Hidden" name="Message" value = "True">
```

الشكل رقم (١٢,٢٠). شفرة صفحة ويب شركة براذهو ماريا.

openkanal.com

تذكر أن أزار الاختيار المستخدمة داخل نماذج الويندوز يجب وضعها داخل إطار أو داخل صندوق تجميع لجعلها مجموعة واحدة من الخيارات، وبذلك نستطيع أن ندرج أكثر من مجموعة داخل النموذج الواحد، أما أزرار اختيار لغة HTML التي تعمل بأسلوب مختلف حيث يسند للصفة NAME لكل مجموعة أزرار اسم واحد، ثم نسند قيمة مختلفة للصفة VALUE لكل زر، ومن ثم نستطيع أن نفرق بين أزرار المجموعة الواحدة، أي أن صفة اسم أزرار الاختيار تستخدم لتصنيف مجموعات الأزرار.

ثم يأتي عنصر القائمة المنسدلة (Select Control) الذي يشبه عنصر القائمة المنسدلة (Combo Box) للغة فيجوال بيسك .نت حيث يحتوي عنصر القائمة المنسدلة على الصفة SIZE التي تحدد عدد الخيارات والتي ستظهر في القائمة، كما يحتوي على علامة التمييز <OPTION> التي تحدد اختيارات القائمة المنسدلة والتي تحتوي على الصفة VALUE التي تحدد قيمة الاختيار مع العلم أن هذه الصفة هي صفة اختيارية.

تمثل أوامر آخر أربعة سطور ثلاثة أزرار وصندوق نص مخفي حيث يستخدم الزر RESET لمسح محتويات عناصر النموذج عدا عنصر القائمة المنسدلة لأن قيمه التي تحتويها قيم ثابتة وتتواجد داخل شفرة HTML، ويستخدم الزر SUBMIT لإرسال محتويات النموذج للملف الموضح اسمه داخل الصفة ACTION، وأخيراً يحتوي صندوق النص المخفي على القيمة True التي ستتغير إلى القيمة False إذا لم تنفذ شفرة عميل الويب قبل الضغط على الزر SUBMIT.

يوضح الشكل رقم (١٢,٢١) برنامج لغة VBScript الذي سينفذ لدى العميل، تمثل لغة VBScript لغة فرعية من لغة Visual Basic .NET، وسوف نتعلم من هذا البرنامج كيف نتعامل مع عنصر داخل نموذج وكيف نسترجع القيمة التي أدخلها المستخدم داخل هذا العنصر. تم فصل أوامر هذا البرنامج بعيداً عن شفرة HTML داخل التركيب Script باستخدام علامة التمييز <SCRIPT> حيث استخدمنا الصفة LANGUAGE لتحديد اللغة المستخدمة (لغة VBScript).

عزيزي القارئ تعالى معي لنتتبع سوياً أوامر تنفيذ هذا البرنامج حيث يبدأ التنفيذ عند حدوث الحدث OnMouseOver على الزر btnCheck الذي سيستدعي الإجراء btnCheck_OnMouseOver الذي يعرض رسالة، ثم يستدعي الإجراء btnCheck_OnClick الذي سيظهر رسائل عن قيم عناصر النموذج. يعرف الإجراء btnCheck_OnClick متغيرات باستخدام الكلمة المحجوزة DIM لتخزين قيم صناديق النص ونافذة النص، وتذكر أن لغة VBScript تمثل لغة فرعية من لغة Visual Basic .NET، ولذلك لا تطلب تعريف أنواع البيانات عند تعريف المتغيرات، كما يمكنك استخدام إجراءات التحويل للحصول على نوع البيان المطلوب، وأخيراً لاحظ أن آخر أمر داخل الإجراء btnCheck_OnClick يقوم بإسناد القيمة False لصندوق النص المخفي.

```
<html>
<head>
<title>Bradshaw Marine Web Survey</title>
</head>
<script language=VBScript>
   ' The following code is for the onclick event
   ' for the Check Button
   sub btnCheck_onclick()
      ' Shows how to reference form controls
      dim fn, ln, txArea
      fn=frmTest.txtFirstName.value     'save first name
      ln=frmTest.txtLastName.value      'save last name
      txArea=frmTest.txtDesire.value      'save text area

      ' Display these text and text area values
      msgbox "First Name is:  " & fn
      msgbox "Last Name is:  " & ln
      msgbox "Desirable features text is:  " & txArea

      ' Display the check boxes if checked
      if frmTest.chkSailBoat.checked then msgbox "Sailboat Experience"
      if frmTest.chkSkiBoat.checked then msgbox "Ski Boat Experience"
      if frmTest.chkCabinCruiser.checked then msgbox "Cabin Cruiser Experience"
      ' Display which radio button checked
      'if frmtest.radio1.checked then msgbox "Monthly"
      select case true
         case frmtest.radio1.checked : msgbox "Prefer monthly payment"
         case frmtest.radio2.checked : msgbox "Prefer semi-annual payment"
         case frmtest.radio3.checked : msgbox "Prefer annual payment"
         case else :msgbox "none"
      end select

      ' Display the check color from the Selection box
      msgbox frmTest.cboCheckColor.value
      if frmTest.cboCheckColor.value = "Blue" then msgbox "This really means
         light blue"
      end if
      '  Example of referencing a hidden value
      frmTest.Message.value = false

   end sub
   ' Example of events for other form controls
   Sub radio1_onclick()
      msgbox "You selected the Monthly option"
   end sub
   sub chkSailBoat_onclick()
      msgbox "You checked Sailboat"
   end sub
   sub btnCheck_OnMouseOver
      msgbox "Mouse over Check Button"
      call btnCheck_onClick()
   end sub
   sub submit
      Call frmTest.submit
   end sub
</script>
```

الشكل رقم (١٢,٢١). شفرة جانب عميل الويب.

توجد ثلاثة إجراءات أخرى وهي: أولاً الإجراء radio1_onclick الذي يتم استدعاؤه عند الضغط على زر الاختيار radio1، وثانياً الإجراء chkSailboat_onclick الذي يتم استدعاؤه عند التأشير على صندوق التأشير chkSailboat، وثالثاً الإجراء submit الذي يستخدم الإجراء Call لإرسال محتويات النموذج إلى الملف Bradshaw_Marine_response.asp (المسند للصفة ACTION).

يوضح الشكل رقم (١٢,٢٢) شفرة أوامر الملف Bradshaw_Marina_response.asp الذي سيتم استدعاؤه ليستقبل بيانات النموذج، وتذكر أن ملفات ASP يجوز أن تحتوي على شفرة ASP بالإضافة إلى شفرة HTML عادياً داخل نفس الملف حيث يتم تمييز أوامر لغة ASP من خلال كتابتها بين العلامتين "<% %>"، وبذلك يتمكن محرك ASP أن يميز هذه الأوامر وينفذها. يحتوي هذا الملف على أمر تحديد نوع لغة البرمجة المستخدمة (اللغة VBScript)، ثم الأمر Option Explicit الذي يجبر المبرمج على توصيف المتغيرات قبل استخدامها، ثم أمر إسناد القيمة True للماكرة الكائن Response (Buffer) التي تتسبب في إرسال نتيجة المعالجة دفعة واحدة في نهاية معالجة الأوامر.

ضبط لغة البرمجة النصية بلغة VBScript

طلب صريح بتعريف متغيرات البرنامج

إرسال جميع طلبات الويب مرة واحدة للخادم

استخلاص بيانات النموذج عن طريق المجموعة Form المعرفة داخل الكائن Request

```
<%@ Language=VBScript %>
<% option explicit %>
<% Response.Buffer=true %>
<HTML>
<Head><Title>Response Page</Title></Head>
<BODY>
        <P><h1> This is the response </h1></P>
<%
dim fn, ln, txArea
'  get posted values from submitted form
'  uses the request object
'  uses request.form because Method=Post on submitted form
'  one would use request.querystring if Method=Get
fn=request.form("txtFirstName")
ln=request.form("txtLastName")
txArea=request.form("txtDesire")

'       write these values using the response object
response.write "<BR>" & "The first name is: " & fn & "<BR>"
response.write "<BR>" & "The last name is: " & ln & "<BR>"
response.write "<BR>" & "Desirable features for the dock:  " & txArea & "<BR>"

' see what check boxes have been checked
if request.form("chkSailBoat")= "sb" then  'note can also use ="on"
        response.write "<BR>" & "You have Sailboat experience" & "<BR>"
end if
if request.form("chkSkiBoat")= "skb" then
        response.write "<BR>" & "You have Ski Boat experience" & "<BR>"
end if
if request.form("chkCabinCruiser")="cc" then
        response.write "<BR>" & "You have Cabin Cruiser experience" & "<BR>"
end if

' see which radio button has been selected
select case ucase(Request.form("rdbType"))
    case "MONTHLY"
       response.write "<BR>" & "You prefer to pay monthly" & "<BR>"
    case "SEMI"
       response.write "<BR>" & "You prefer to pay semi-annually" & "<BR>"
    case "ANNUALLY"
       response.write "<BR>" & "You prefer to pay annually" & "<BR>"
    case else
    response.write "<BR>" & "You didn't select a preferred payment method" &
"<BR>"
end select

'  see which value in the combo box has been selected
response.write "<BR>" & " Boat color is " & request.form("cboCheckColor")

response.write "<BR>" & "The hidden value is: " & request.form("Message")

%>

</BODY>
</HTML>
```

الشكل رقم (١٢,٢٢). شفرة Bradshaw_Marina_response.asp.

Obeikandl.com

يحتوي ملف ASP أولاً على شفرة HTML تحديد عنوان الصفحة ("Response Page") وعلى الفقرة التي تحتوي على العبارة "This is the response page"، ثم تأتي شفرة ASP التي تبدأ بتعريف ثلاثة متغيرات لاستقبال بيانات النموذج، وبما أن النموذج تم إرساله بأسلوب POST، إذن سيتم استقبال البيانات من المجموعة Form التي تنتمي إلى الكائن Request باستخدام أسماء عناصر النموذج هكذا:

```
<HTML>
<Head><Title>Response Page</Title></Head>
<BODY>
        <P><h3> This is the response </h3></P>
<%
dim fn, ln, txArea
'  get posted values from submitted form
'  uses the request object
'  uses request.form because Method=Post on submitted form
'  one would use request.querystring if Method=Get
fn=request.form("txtFirstName")
ln=request.form("txtLastName")
txArea=request.form("txtDesire")
```

لقد استخدمنا في هذا المثال الإجراء Write الذي ينتمي للكائن Response لإرسال رسائل لعميل الويب، ولاحظ أنه يمكن إرسال شفرة HTML مع نصوص الرسائل لتنسيق النص بوضع علامات تمييز لغة HTML بين علامتي تنصيص (مثل "<BR>" التي تستخدم لطباعة سطر جديد)، وترسل الأوامر التالية بيانات النموذج ثانيةً إلى الخادم.

```
'       write these value using the response object
response.write "<BR>" & "The first name is: " & fn & "<BR>"
response.write "<BR>" & "The last name is: " & ln & "<BR>"
response.write "<BR>" & "Desirable features for the dock:  " & txArea &
"<BR>"
```

بعد ذلك نكتب أوامر اختبار لمعرفة ما هي صناديق التأشير التي اختارها المستخدم، ومن ثم نعرف أي المراكب التي تم اختيارها، وبذلك نرسل له رسائل بالمراكب التي هو خبير بها، ولاحظ شروط اختيار أمر If الذي يستخدم قيمة صندوق التأشير التي أسندناها له عند تصميم النموذج حيث يعيد الشرط القيمة True إذا احتوى صندوق التأشير على هذه القيمة في حال التأشير عليه، ولاحظ أيضاً أنه يمكننا استخدم القيمة On بدلاً من قيمته.

```
' see what check boxes have been checked
   if request.form("chkSailBoat")= "sb" then  'note can also use ="on"
      response.write "<BR>" & "You have Sailboat experience" & "<BR>"
end if
if request.form("chkSkiBoat")= "skb" then
   response.write "<BR>" & "You have Ski Boat experience" & "<BR>"
end if
If request.form ("chkCabinCruiser")="cc" then
    response.write "<BR>" & "You have Cabin Cruiser experience" & "<BR>"
end if
```

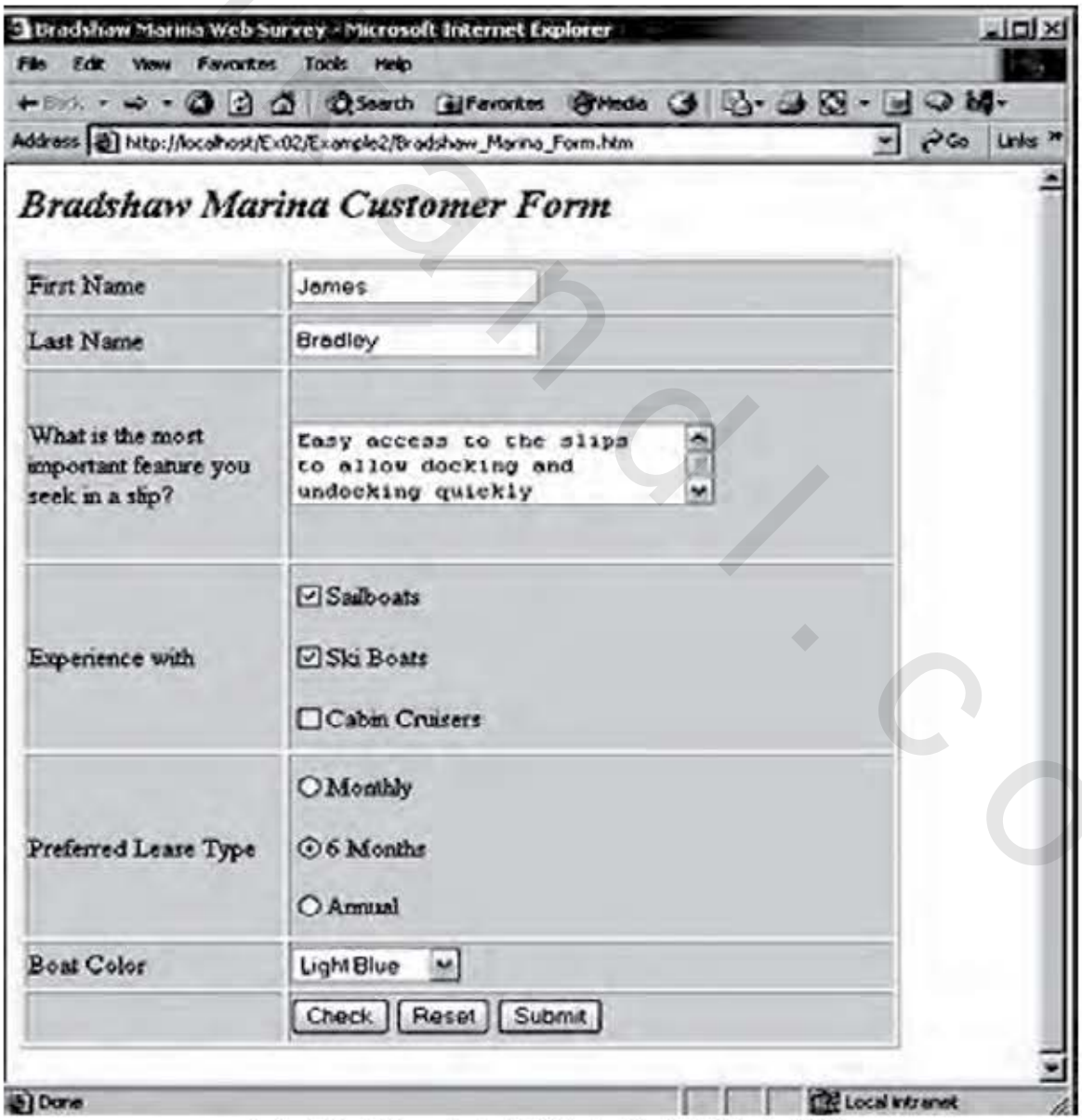
ثم نختبر قيمة زر الاختيار لمعرفة أي أزرار الاختيار تم اختيارها، ومن ثم نعرف أي أسلوب دفْع يفضله المستخدم، ثم نرسل له رسائل بطريقة الدفع المفضلة لديه، ولاحظ استخدام أمر Select Case الذي يقارن القيمة الحالية لزر الاختيار بقيم أزرار الاختيار المدخلة أثناء تصميم النموذج، ولاحظ أيضاً استخدام اسم زر الاختيار rdbType نفسه لجميع أزرار الاختيار عند تصميم النموذج لجعلهم مجموعة واحدة.

```
'see which value in the combo box has been selected
select case ucase(Request.Form("rdbType"))
   case "MONTHLY"
      response.write "<BR>" & "You prefer to pay monthly" & "<BR>"
   case "SEMI"
      response.write "<BR>" & "You prefer to pay semi-annually" & "<BR>"
   case "ANNUALLY"
      response.write "<BR>" & "You prefer to pay annually" & "<BR>"
   case else
   response.write "<BR>" & "You didn't select a prefered pay method" &
"<BR>"
end select
```

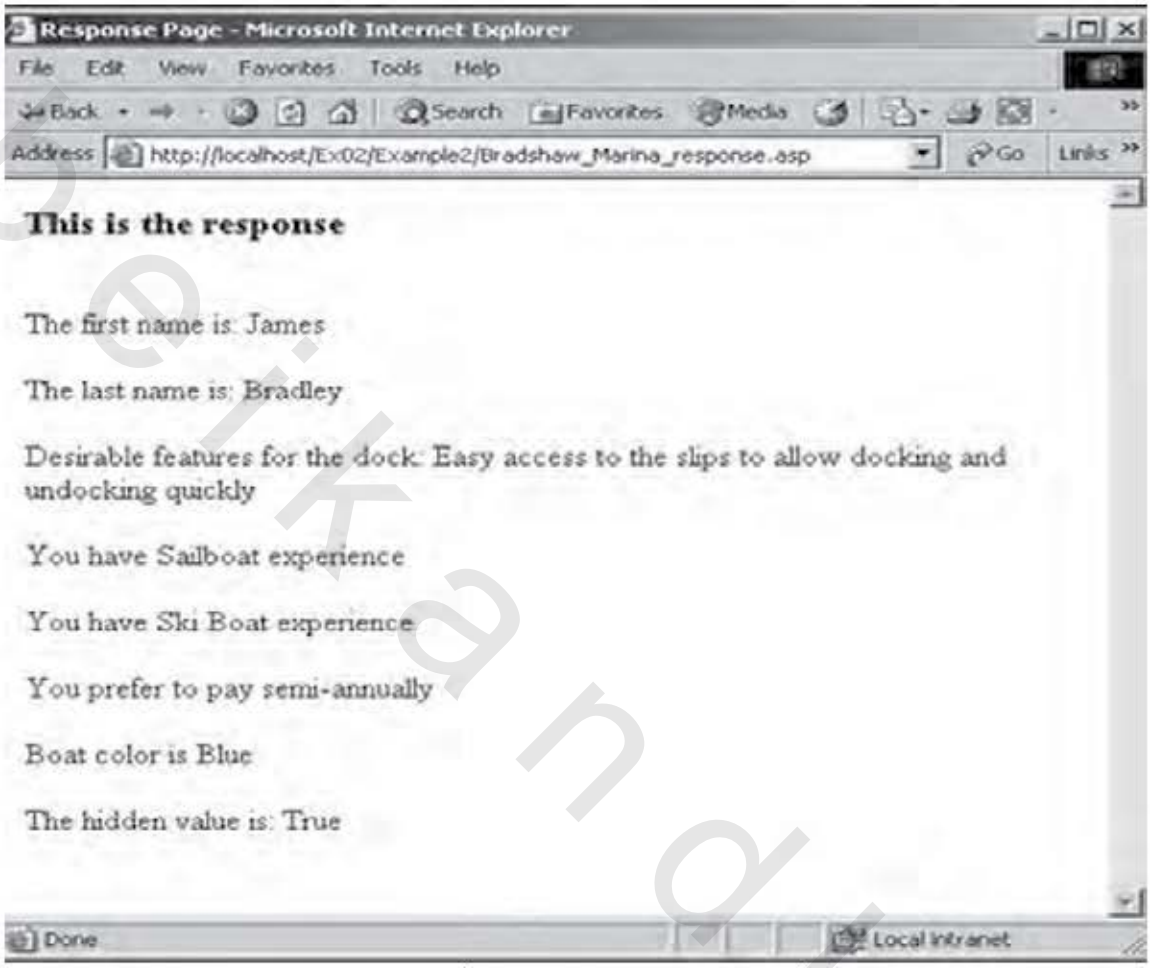
وأخيراً نحدد رسالة باللون الذي اختاره المستخدم من العنصر Select (القائمة المنسدلة) ونظهرها، وأيضاً نقوم بعرض رسالة عن قيمة صندوق النص المخفي، ثم نغلق شفرة ASP باستخدام العلامة %>، ثم نغلق شفرة HTML بكل من علامتي </BODY> و </HTML>.

```
'  see which value in the combo box has been selected
response.write "<BR>" & " Boat color is " & request.form("cboCheckColor")
response.write "<BR>"
response.write "<BR>" & "The hidden value is: " & request.form("Message")

%>
</BODY>
</HTML>
```

وبشكل عام، فإن تقنية ASP تمكّنك من دمج أوامر نصية (Scripts) داخل شفرة HTML والتي يجب أن تخزن سوياً في ملف ASP الذي يتم معالجته بواسطة محرك ASP المتواجد لدى خادم الويب، ثم إرسال ناتج المعالجة إلى عميل الويب ثانياً، أما فكرة دمج شفرة كل من ASP مع شفرة HTML فإنها تقدم تقنية قوية لتطوير صفحات الويب متغيرة المحتوى، وسوف نرى في الموضوع التالي أن تقنية فيجوال ستوديو نت مازالت تتوافق مع تقنية ASP، ومن ثم نستطيع استخدام تقنية ASP.NET لتطوير تطبيقات الويب. يوضح الشكل رقم (١٢,٢٣) مثالاً لنموذج يحتوي على بيانات أحد عملاء شركة برادشو مارينا، كما يعرض الشكل رقم (١٢,٢٤) المخرجات المناظرة لهذا النموذج عند إرساله.



الشكل رقم (١٢,٢٣). النموذج الكامل لصفحة ويب شركة برادشو مارينا.



الشكل رقم (١٢,٢٤). صفحة استجابة شركة برادشو مارينا.

استخدام تقنية ASP.NET لتطوير صفحات الويب
Using ASP.NET to Develop Web Pages

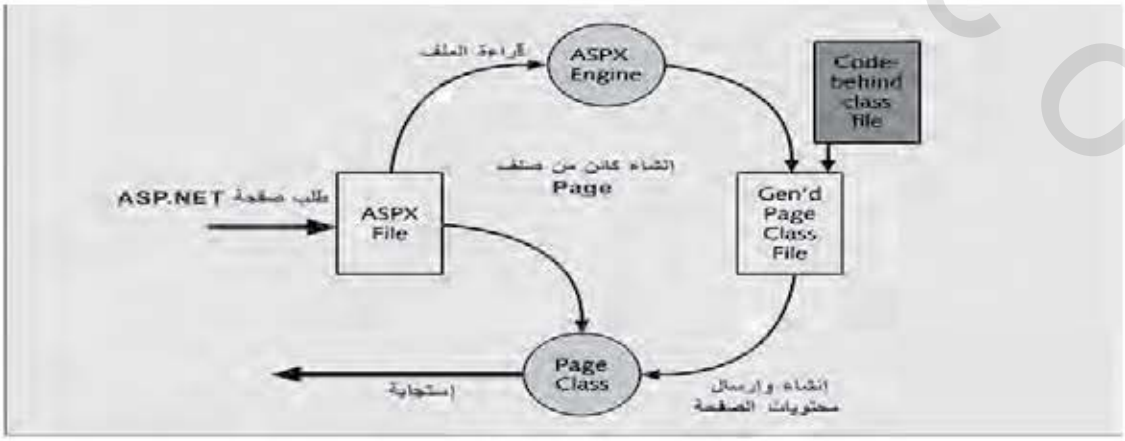
إن هدف شركة مايكروسوفت من تقديم كل من إطار عمل .نت وفيجوال ستوديو .نت هو تقديم بيئة تطوير متكاملة تدعم تطوير كل من تطبيقات الويندوز وتطبيقات الويب، ولقد رأينا في الفصل العاشر كيف نطور تطبيقات ويندوز باستخدام لغة فيجوال بيسك .نت، وقد عرضنا في الفقرة السابقة من هذا الفصل المفاهيم الأساسية لتطوير تطبيقات الويب، وسوف نتحدث في بقية هذا الفصل عن استخدام تقنية ASP لتطوير تطبيقات الويب باستخدام أيضاً لغة فيجوال بيسك .نت، ولأن تطوير تطبيقات الويب باستخدام تقنية ASP تشبه تطوير تطبيقات الويندوز، إذن ستستفيد عزيزي القارئ من خبرتك السابقة، ولكن على أي حال يجب علينا أولاً معرفة المفاهيم الجديدة التي قدمتها تقنية ASP مثل مفهوم التطوير باستخدام أسلوب "Drag & Drop" وميزة ترجمة شفرة الأوامر بشكل تلقائي (Dynamic Compilation) لتنفيذ أسرع وميزة فصل شفرة البرنامج عن شفرة محتويات وميزة

عناصر التحقق من صحة البيانات (Validation Controls). سوف نناقش هذه المفاهيم باختصار قبل تطوير تطبيق الويب باستخدام تقنية ASP.

فصل شفرة البرنامج عن المحتويات
Separating Code form Content

إن أهم ميزة قدمتها تقنية ASP.NET هو فصل شفرة البرنامج ASP.NET عن شفرة محتويات واجهة الصفحة، وتذكر أن صفحات ASP تحتوي على كل من شفرة ASP مع شفرة HTML في نفس الملف حيث إنه من الصعب جداً الفصل بينهما. استخدمت تقنية ASP.NET أسلوب Code-behind الذي يشبه مفهوم البرمجة المعتمدة على الأحداث Event-driven Programming والمتبعة في تطوير تطبيقات الويندوز. عند تطوير صفحة ويب باستخدام تقنية ASP.NET يتم إنشاء ملفين. يحتوي الملف الأول على محتويات واجهة الصفحة (شفرة HTML) ويأخذ الامتداد aspx.، ويحتوي الملف الثاني على شفرة ASP.NET (الإجراءات التي ستنفذ نتيجة أحداث عناصر صفحة الويب) ويأخذ الامتداد aspx.vb.. ولإضافة إجراءات معالجة حدث عنصر ما داخل الصفحة نقوم بالضغط المزدوج على هذا العنصر، فعندئذ ستظهر نافذة كتابة الأوامر حيث يظهر إجراء معالجة الحدث الافتراضي لهذا العنصر بشكل تلقائي، وكما ذكرنا سابقاً سوف تحفظ تقنية ASP.NET بهذه الإجراءات داخل ملف مستقل (الملف الثاني ذات الامتداد aspx.vb).

يوضح الشكل رقم (١٢,٢٥) ماذا يحدث عندما يطلب المستخدم صفحة ويب من خادم الويب. سيتم قراءة هذا الملف باستخدام برنامج محرك ASPX الذي ينتج ملف الصنف Page والذي يترجم بواسطة المكون CLR قبل إرسال المخرجات، ثم يدمج ملف Code-behind المترجم والذي يحتوي على إجراءات معالجة أحداث هذه الصفحة مع الصنف Page، ثم يتم تخزين شفرة صنف هذه الصفحة على القرص الصلب لطلبات المستخدم القادمة، ومن ثم لن تحدث الترجمة مرة ثانية إلا إذا حدث تغيير في شفرة أوامر الصفحة.



الشكل رقم (١٢,٢٥). ترجمة صفحة ويب ASP.NET.

استخدام عناصر HTML وعناصر HTML Server وعناصر ASP.NET
Using HTML, HTML Server Controls, and ASP.NET Server Controls

نستطيع أن نطور تطبيقات ويب تتكون من كل من عناصر HTML وعناصر HTML تعمل لدى خادم الويب (HTML Server Controls) وعناصر ASP.NET التي تعمل على خادم الويب (ASP.NET Server Controls)، ولقد صممنا سابقاً صفحات ويب باستخدام عناصر HTML، أما كل من عناصر HTML وعناصر ASP.NET اللذان يعملان على خادم الويب فيتشابه عملهما إلى حدٍّ كبير، إلا أنهما يختلفان في خصائصهما، فمثلاً نقوم باسترجاع قيمة عنصر صندوق نص النوع الأول باستخدام الصفة VALUE، بينما نستخدم الصفة TEXT لاسترجاع قيمة عنصر صندوق نص النوع الثاني (مثل عناصر تطبيقات الويندوز)، كما أن عناصر ASP.NET التي تعمل على خادم الويب تحتوي على عناصر أكثر مثل العنصر AdRotator والعنصر Calendar غير المتواجدين داخل عناصر HTML التي تعمل على خادم الويب.

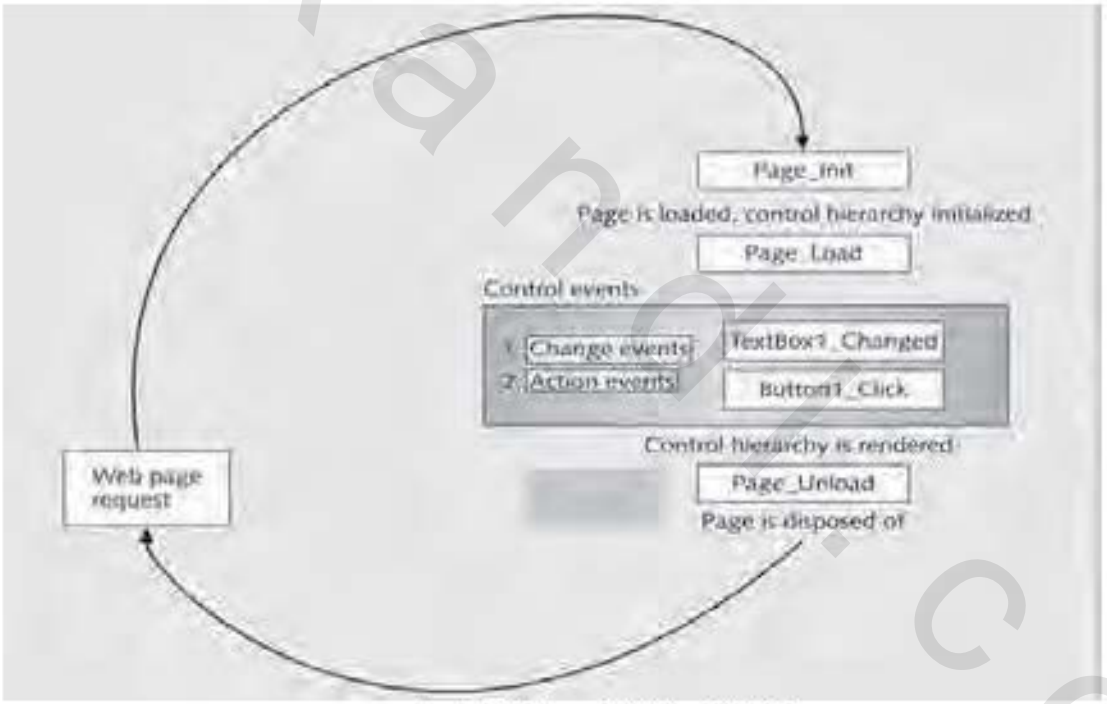
تحتوي عناصر ASP.NET التي تعمل لدى خادم الويب على صفات وإجراءات وأحداث مثل عناصر تطبيقات الويندوز، كما أن معظم هذه العناصر تدعم مفهوم "Data Binding" الذي يربط عناصر النموذج بمصدر بيانات (مثل جدول في قاعدة بيانات)، فمثلاً إن كلاً من العنصر DataGrid والعنصر DataList والعنصر Repeater يتم استخدامها فقط لعرض بيانات داخل النموذج عند ارتباطها بمصدر للبيانات. تتميز عناصر ASP.NET التي تعمل لدى خادم الويب أيضاً باستنتاج نوع متصفح الويب، ومن ثم تنتج عناصر صفحة الويب بناء على نوع المتصفح.

وأخيراً تحتوي عناصر ASP.NET التي تعمل لدى خادم الويب أيضاً على مجموعة هامة من العناصر تستخدم للتحقق من صحة البيانات قبل إرسال محتويات النموذج إلى خادم الويب، ويتم وضع عناصر التحقق من صحة البيانات على النموذج مثل بقية عناصر ASP.NET التي تعمل لدى خادم الويب، كما تحتوي مجموعة عناصر التحقق من صحة البيانات على عنصر تلخيص الأخطاء الذي يظهر جميع أخطاء إدخال البيانات في موضع واحد من الصفحة.

دورة حياة أحداث صفحة ASP.NET
Understanding the ASP.NET Page Event Life Cycle

يحتوي صنف صفحة ASP.NET (الصنف Page) على خصائص وأحداث وإجراءات حيث يوضح الشكل رقم (١٢,٢٦) على دورة حياة أحداث هذا الصنف والتي تبدأ بالحدث Page_Init وذلك عندما يطلب عميل الويب صفحة الويب، مع العلم أن هذا الحدث يحدث قبل تحميل عناصر النموذج. ربما يكون اختيار وضع شفرة الإعداد والاتصال بقاعدة بيانات التطبيق داخل إجراء الحدث Page_Init يمثل اختياراً جيداً، حيث يتم إنتاج هذا الإجراء آلياً بواسطة مصمم نماذج الويب؛ ولذلك يجب توسعة هذا الجزء من الأوامر (كما هو واضح في المثال التالي) والبحث عن هذا الإجراء ثم كتابة أوامر تهيئة الاتصال بقاعدة البيانات وذلك بعد الأمر InitializeComponent().

```
#Region " Web Form Designer Generated Code "

    'This call is required by the Web Form Designer.
    <System.Diagnostics.DebuggerStepThrough()> Private Sub
InitializeComponent()

    End Sub

    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()
    End Sub

#End Region
```



الشكل رقم (١٢,٢٦). دورة حياة صفحة ASP.NET.

ثم يحدث الحدث Page_Load بعد حدوث الحدث Page_Init الذي يعطي فرصة جيدة لكتابة أوامر فحص قيمة الصفة Postpack وكتابة الأوامر المناسبة بناء على قيمتها حيث تأخذ الصفة Postpack القيمة False عند استدعاء صفحة الويب الحالية للمرة الأولى وتأخذ القيمة True عند استدعائها في المرات التالية، ومن ثم يمكن كتابة أوامر إعداد عناصر الصفحة عند تحميل الصفحة للمرة الأولى فقط (مثل تعبئة عنصر القائمة المنسدلة من قاعدة البيانات) وذلك في حالة أن الصفة Postpack تساوي False كما هو واضح في أوامر المثال التالي.

```
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
      'Put code here for first time page loads
   End If
End sub
```

ثم يحدث الحدث Page_Unload الذي يمكّن المبرمج من كتابة أوامر إنهاء المعالجة قبل إلغاء تحميل الصفحة مثل إغلاق اتصال بقاعدة البيانات أو إغلاق ملفات مفتوحة أو تدمير كائنات لا يوجد لها احتياج، مع العلم أن هذا الحدث يحدث بعد إلغاء تحميل عناصر الصفحة ولكن قبل إرسال الصفحة إلى عميل الويب.

إنشاء تطبيق ويب باستخدام تقنية ASP.NET
Creating an ASP.NET Web Application

لكي تنشئ تطبيق ويب باستخدام تقنية ASP.NET، قم بإنشاء مجلد (افتراضي) أولاً، ثم شغل برنامج فيجوال ستوديو .نت، ولعمل ذلك اتبع الخطوات التالية:

١- ابحث عن المجلد Ex3 داخل المجلد Chap12\Examples داخل ملفات هذا الكتاب المخصص للطلاب، ثم انسخه إلى المجلد Chap12\Exercises في جهاز الحاسب الآلي الخاص بك (أنشئ هذا المجلد إذا لم يكن موجود بالفعل).

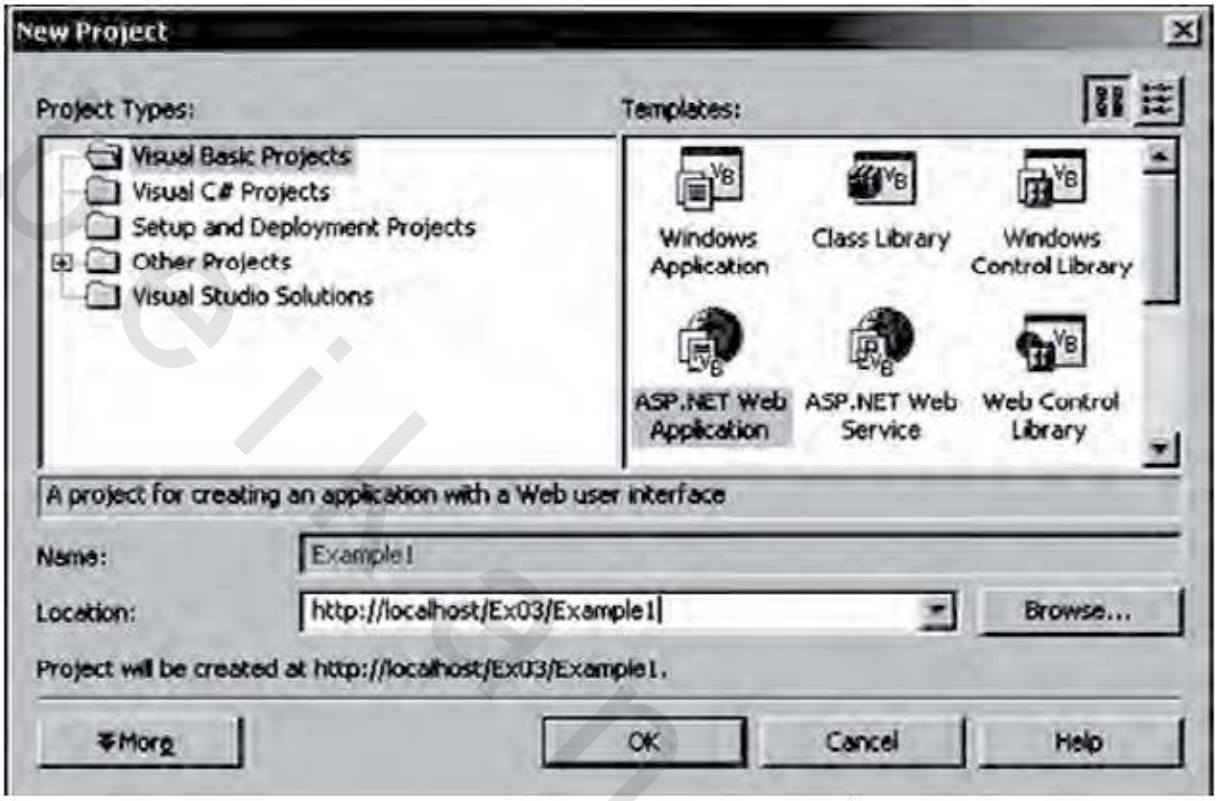
٢- أنشئ مجلداً افتراضياً يسمى Ex03 (راجع موضوع التعامل مع مجلدات الويب لمعرفة خطوات إنشاء المجلدات الافتراضية).

٣- شغل برنامج فيجوال ستوديو .نت بالضغط على الزر Start ثم Programs (أو All Programs داخل ويندوز XP)، ثم شاور على Microsoft Visual Studio .NET، ثم اضغط على Microsoft Visual Studio .NET، فعندئذ ستظهر صفحة Start Page.

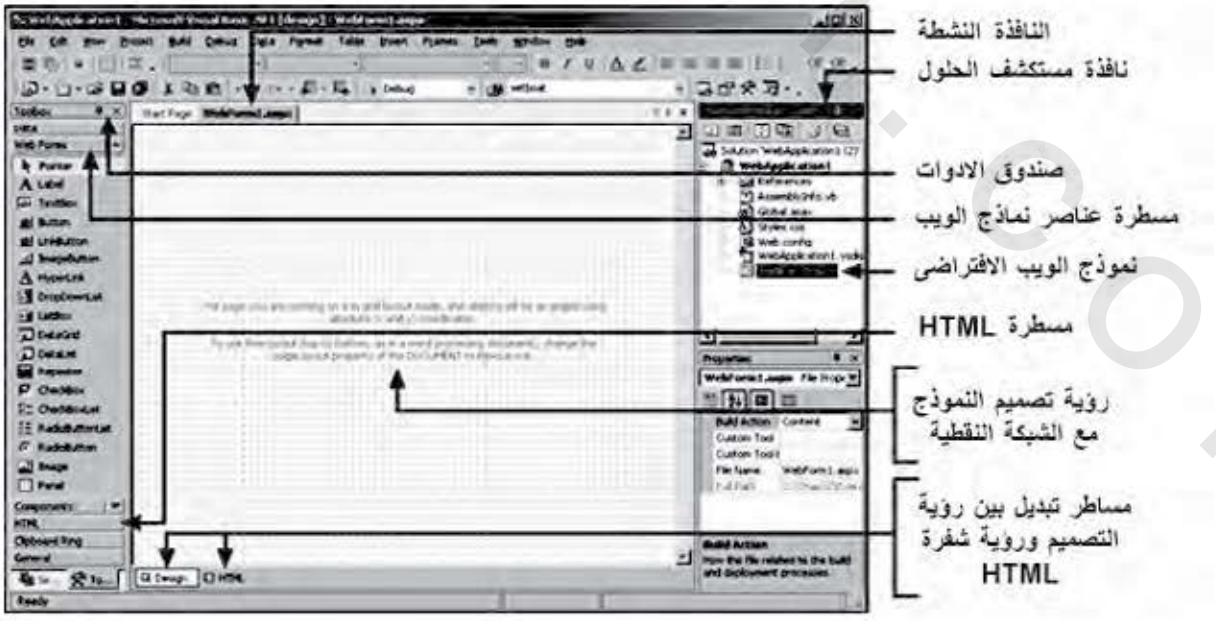
٤- اضغط على الزر New Project داخل صفحة Start Page لفتح نافذة New Project، ثم اضغط على Visual Basic .NET داخل قائمة Project Types، ثم اختر رمز "ASP.NET Web Applications" من قائمة Templates (انظر الشكل رقم ١٢,٢٧).

٥- إذا كانت هذه هي أول مرة لك تقوم بتطوير تطبيق ويب فسوف يكون المسار الافتراضي للمشروع داخل صندوق النص Location هو "http://localhost" واسم المشروع الافتراضي هو WebApplication1. قم بتغيير المسار ليكون "http://localhost/Ex03"، ثم اضغط على الزر OK، فعندئذ ستظهر نافذة مصمم النماذج كما هو واضح في الشكل رقم (١٢,٢٨).

obeikandl.com



الشكل رقم (١٢,٢٧). نافذة إنشاء مشروع ASP.NET جديد.



الشكل رقم (١٢,٢٨). نافذة مصمم نماذج صفحات ASP.NET.

توضح نافذة مستكشف الحلول أن اسم النموذج الحالي هو "WebForm1.aspx" حيث يمكن تصميم النموذج باستخدام نوعين من التنسيق. أولاً، النوع Grid Layout (الشبكة النقطية) الذي يمثل النوع الافتراضي والذي يمكن تغييره إلى النوع الثاني Page Layout من خلال نافذة الخصائص، كما يوجد هناك رؤيتان لنافذة مصمم النماذج (رؤية التصميم ورؤية شفرة HTML المناظرة للتصميم) حيث يمكن التبديل بينهما من خلال مسطرة Design ومسطرة HTML المتواجدين في الركن الأيسر السفلي للنافذة، مع العلم أن الرؤية الافتراضية هي رؤية التصميم.

وكما هو واضح من الشكل رقم (١٢,٢٨)، فإن نافذة صندوق الأدوات تظهر عناصر المسطرة Web Forms التي تحتوي على عناصر ASP.NET والتي تعمل لدى خادم الويب، ولاحظ أيضاً أن نافذة صندوق الأدوات تحتوي على مسطرة HTML التي تحتوي على عناصر HTML والتي يمكن أن تعمل لدى الخادم أو تعمل لدى العميل فقط.

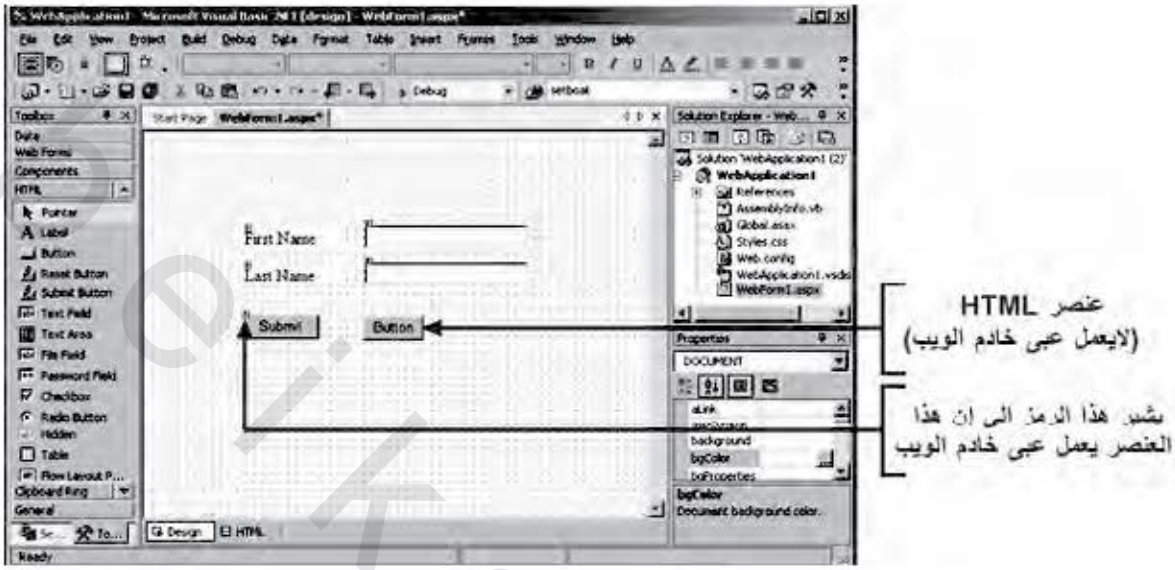
يمكن وضع عنصر النموذج على صفحة الويب بطريقتين وهما: الضغط المزدوج على العنصر أو اختيار العنصر ثم سحبه ووضعه على الصفحة. ولإنشاء عنصر HTML ليعمل لدى خادم الويب فقط، اضغط ضغطاً مزدوجاً على العنصر المراد في نافذة صندوق الأدوات، ثم قم بضغط الزر الأيمن للفأرة عليه، ثم اختر Run At Server (يمكنك وضع العنصر في أي مكان داخل النموذج بسحبه ووضعه في المكان المناسب).

يوضح الشكل رقم (١٢,٢٩) نموذجاً يحتوي على خمسة عناصر تعمل لدى خادم الويب وعلى عنصر HTML واحد، مع العلم أن ASP.NET تضع علامة في الركن الأيسر العلوي للعنصر الذي يعمل لدى الخادم، ومن ثم يمكنك تمييز العناصر التي تعمل لدى الخادم وعناصر HTML فقط، والآن يمكنك تشغيل التطبيق بالضغط على زر التشغيل الموجود في المسطرة العلوية.

يسند لكل عنصر ASP.NET يعمل لدى الخادم اسم وحيد، مع العلم أن شفرة واجهة الصفحة ومحتوياتها تخزن داخل ملف HTML الذي يشمل أيضاً على علامات تميز عناصر ASP.NET. سوف نستخدم في الخطوات التالية خمسة عناصر ASP.NET وعنصر HTML كما هو واضح في الشكل رقم (١٢,٢٩)، ثم نتعرف على الشفرة المناظرة لها. لكي تضع عناصر على صفحة الويب وترى الشفرة المناظرة لها اتبع التالي:

١- اضغط على زر Toolbox لإظهار نافذة صندوق الأدوات إذا لم تكن ظاهرة، ثم اضغط على مسطرة Web Forms داخل نافذة صندوق الأدوات لإظهار عناصر مسطرة Web Form.

٢- اضغط ضغطاً مزدوجاً على العنصر Label داخل نافذة صندوق الأدوات، ثم اسحبه إلى منتصف صفحة الويب كما هو واضح في الشكل رقم (١٢,٢٩)، ثم اكتب First Name داخل الخاصية Text داخل نافذة الخصائص، وبالمثل أضف العنوان Last Name الموضح في الشكل رقم (١٢,٢٩).



الشكل رقم (١٢,٢٩). نموذج ويب يحتوي على كل من عناصر خادم الويب وعناصر HTML.

٣- أضف صندوقي نص إلى صفحة الويب كما واضح في الشكل رقم (١٢,٢٩).

٤- اضغط ضغطاً مزدوجاً على عنصر الزر Button، ثم ضع الزر أسفل العنوان Last Name، ثم غير قيمة الخاصية Text للزر داخل نافذة الخصائص إلى القيمة Submit.

٥- اختر مسطرة HTML في صندوق الأدوات، ثم اضغط ضغطاً مزدوجاً على عنصر الزر، ثم اسحب الزر إلى يمين الزر Submit، ولاحظ أن نص الزر يأخذ القيمة "Button"، وأن تقنية ASP.NET تميز العناصر التي تعمل لدى الخادم بعلامة الركن الأيسر العلوي للعناصر كما هو واضح في الشكل رقم (١٢,٢٩)، ولاحظ أيضاً أن عناصر HTML المضافة مؤخراً لا تعمل لدى الخادم.

٦- اضغط على المسطرة HTML في نافذة مصمم النماذج لرؤية شفرة HTML المناظرة لهذا التصميم، وعندئذ يجب أن تحصل على شفرة مماثلة إلى الشكل رقم (١٢,٣٠). لاحظ أن ASP.NET تظهر الأوامر الموجهة باللون الأصفر التي تظهر في الشكل رقم (١٢,٣٠) ذات خلفية مظللة.

يمثل أول أمر داخل الشكل رقم (١٢,٣٠) أمراً موجهاً (المحاط بعلامتي التمييز <% %>) الذي أسند القيمة VB للصفة Language وأسند القيمة webform1.aspx.vb (الملف الذي سيحتوي على الإجراءات التي ستنفذ استجابة للأحداث) للصفة Code-behind، ثم يأتي تعليق HTML ثم علامات تمييز Head وعلامات تمييز Title، مع العلم أن هذه السطور الأربعة تمثل توثيقاً لصفحة الويب الخالية.

```
<%@ Page Language="vb" AutoEventWireup="false" Codebehind="WebForm1.aspx.vb"
Inherits="Ex03.WebForm3"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
   <HEAD>
      <title>WebForm1</title>
      <meta content="Microsoft Visual Studio.NET 7.0" name="GENERATOR">
      <meta content="Visual Basic 7.0" name="CODE_LANGUAGE">
      <meta content="JavaScript" name="vs_defaultClientScript">
      <meta content=http://schemas.microsoft.com/intellisense/ie5
         name="vs_targetSchema">
   </HEAD>                                                    ASP.NET server controls
   <body MS_POSITIONING="GridLayout">
      <form id="Form1" method="post" runat="server">
         <asp:Label id="Label1" style="Z-INDEX: 101; LEFT: 46px; POSITION:
            absolute; TOP: 139px" runat="server" Width="68px">First Name
         </asp:Label>
         <INPUT id="Button2" style="Z-INDEX: 106; LEFT: 148px; POSITION:
            absolute; TOP: 218px" type="button" value="Button" name="Button2">
         <asp:Button id="Button1" style="Z-INDEX: 105; LEFT: 58px; POSITION:
            absolute; TOP: 219px" runat="server" Text="Submit">
         </asp:Button>
         <asp:TextBox id="TextBox2" style="Z-INDEX: 104; LEFT: 148px;
            POSITION: absolute; TOP: 167px" runat="server">
         </asp:TextBox>
         <asp:Label id="Label2" style="Z-INDEX: 103; LEFT: 46px; POSITION:
            absolute; TOP: 171px" runat="server" Width="68px">Last Name
         </asp:Label>
         <asp:TextBox id="TextBox1" style="Z-INDEX: 102; LEFT: 147px;
            POSITION: absolute; TOP: 137px" runat="server"></asp:TextBox>
      </form>
   </body>
</HTML>
```

الشكل رقم (١٢,٣٠). رؤية HTML لصفحات الويب.

والآن قم بالبحث عن عنصر العنوان (Label) في النموذج الذي يعرف بعلامتي التمييز <asp:Label>...</asp:Label>. تشبه علامات تمييز عناصر ASP.NET علامات HTML مع بعض الاختلاف، وتستخدم عناصر ASP.NET الصفة ID بدلاً من الصفة NAME، مع العلم أن هذه الخصائص توجد داخل نافذة الخصائص حيث يمكن رؤيتها وتعديلها بشكل أسهل من رؤيتها وتعديلها داخل شفرة HTML مباشرة. تحتوي جميع العناصر التي تعمل لدى الخادم الصفة runat="server" حيث سيتم معالجة هذه العناصر لدى الخادم، وإذا بحثت عن عنصر HTML الوحيد (العنصر Input) فلن تجد أي صفة تشير إلى أنه يجب أن يعالج لدى الخادم، وعلى أي حال إذا أردت تحويل هذا العنصر إلى عنصر يعمل لدى الخادم فاضغط عليه بالزر الأيمن للفأرة في رؤية التصميم، ثم اختر Run As Server Control، فعندئذ ستضاف الصفة runat="server" إلى شفرة HTML الخاصة به.

والسؤال الذي يجب أن يطرح نفسه في هذا المقام هو متى نستخدم عناصر HTML وعناصر HTML التي تعمل لدى الخادم وعناصر ASP.NET التي تعمل لدى الخادم؟ سوف نستخدم عناصر HTML وعناصر ASP.NET

في معظم تطبيقات الويب، ولأن العناصر التي تعمل لدى الخادم تحتاج إلى إعادة طلب صفحة الويب من الخادم للمعالجة فإن العناصر التي لا تحتاج إلى معالجة لدى الخادم مثل العناوين يفضل أن تكون من النوع HTML، أما الفرق بين عناصر ASP.NET التي تعمل لدى الخادم وعناصر HTML التي تعمل لدى الخادم غير واضح، وبما أن عناصر ASP.NET التي تعمل لدى الخادم تشمل على مجموعة أغنى من العناصر (مثل عنصر التقويم وعنصر التحقق من صحة البيانات)، فإن من المفضل التركيز على استخدامها في تطوير تطبيقات الويب.

عزيزي القارئ لديك مطلق الحرية في كيفية تصميم نماذج الويب وذلك باستخدام رؤية التصميم بواسطة إضافة العناصر إلى النموذج أو كتابة شفرة الأوامر داخل رؤية HTML، ولكن إن معظم مطوري التطبيقات يستخدمون رؤية التصميم للعمل مع عناصر نماذج الويب ثم ينتقلون إلى رؤية HTML لإدخال بعض التحسينات الطفيفة على شفرة HTML الخاصة بالعناصر.

إنشاء تطبيق ASP.NET
Creating an ASP.NET Survey Application

عزيزي القارئ لقد أصبح لديك الآن فكرة جيدة عن تصميم نماذج الويب باستخدام تقنية ASP.NET حيث تستطيع تطوير تطبيق ويب لشركة برادشو مارينا لجمع معلومات حول عملاء الشركة حيث نحصل على المعلومات من خلال نموذج ويب، ثم نمرر هذه المعلومات إلى نموذج ويب آخر لمشاهدتها فقط وذلك على عكس التطبيقات الواقعية التي تخزن المعلومات بعد الحصول عليها في قاعدة البيانات (سوف نتعرض لموضوع قاعدة البيانات والكائنات المستمرة لاحقاً في هذا الكتاب)، كما سيعرض هذا المثال مفاهيم أخرى هامة عن تطوير تطبيقات الويب باستخدام تقنية ASP.NET. يوضح الشكل رقم (١٢,٣١) النموذج الأول الذي سيستخدم لجمع المعلومات من العميل.

يظهر في قمة النموذج عنصران من النوع ASP.NET Image لعرض صورتين لمركبين شراعيين، كما يظهر بين الصورتين عنوان الشركة داخل عنوان (Bradshaw Marina)، ولكي تظهر عناصر هذا النموذج بهذا التنسيق استخدمنا جدولاً لكي يحتوي على العناصر بين خلاياه حيث أدرجنا جدولاً ثم ضبطنا جميع صفوفه وأعمدته داخل صفحة ويب ثم ضبطنا الصفة CELLPADDING بالقيمة خمسة، ثم أضفنا عناصر النموذج داخل الجدول مع العلم أن جميع العناصر من النوع ASP.NET التي تعمل لدى الخادم عدا عنصر الزر Reset الذي ينتمي إلى عناصر HTML، وربما نحتاج أن نراجع شفرة HTML الناتجة وإزالة بعض علامات التمييز مثل علامة الفقرة لضبط المسافة بين العناصر.

الشكل رقم (١٢,٣١). صفحة شركة براندشو مارينا باستخدام ASP.NET.

تحتوي أول خمسة صفوف للعمود الأيسر على خمسة عناصر عناوين من النوع ASP.NET التي تعمل لدى الخادم ويحتوي الصف السادس للعمود الأيسر على زرين. ينتمي الزر Submit إلى عناصر ASP.NET وينتمي الزر Reset إلى عناصر HTML، وسوف نناقش شفرة الزر بعد مناقشة عناصر النموذج.

يحتوي العمود الثاني من الجدول على أشهر عناصر تستخدم لجميع المعلومات حيث يتم إسناد اسم ذو معنى للصفة ID لجميع هذه العناصر، وتنتمي العناصر الثلاثة الأولى على صناديق النص ASP.NET التي تستخدم للحصول على كل من الاسم الأول والاسم الأخير للعميل وعلى أهم معالم يريدها العميل في المرسى (لاحظ ضبط صفة TextMode للعنصر الثالث بالقيمة Multiline)، ثم تأتي صناديق التأشير التي تستخدم لجمع معلومات عن خبرة العميل بالمركب حيث يتعلم العميل أن يؤشر على أكثر من خيار ثم تأتي أزرار الاختيار التي تمكّن العميل من اختيار أسلوب الدفع المناسب حيث يستطيع العميل أن يؤشر على أكثر من خيار، ثم تأتي أزرار الاختيار التي تمكّن العميل من اختيار أسلوب الدفع المناسب حيث لا يستطيع العميل أن يختار أكثر من خيار ولذلك يفضل استخدام أزرار الاختيار، ثم تأتي قائمة الاختيار المنسدلة لتمكّن العميل من اختيار لون المركب حيث يمكننا

إضافة قيم ألوان للقائمة المنسدلة مثلما نفعل مع القائمة المنسدلة داخل تطبيقات الويندوز أثناء وقت تصميم النموذج وذلك بإدراج قيم داخل المجموعة (أحد خصائص عنصر القائمة المنسدلة).

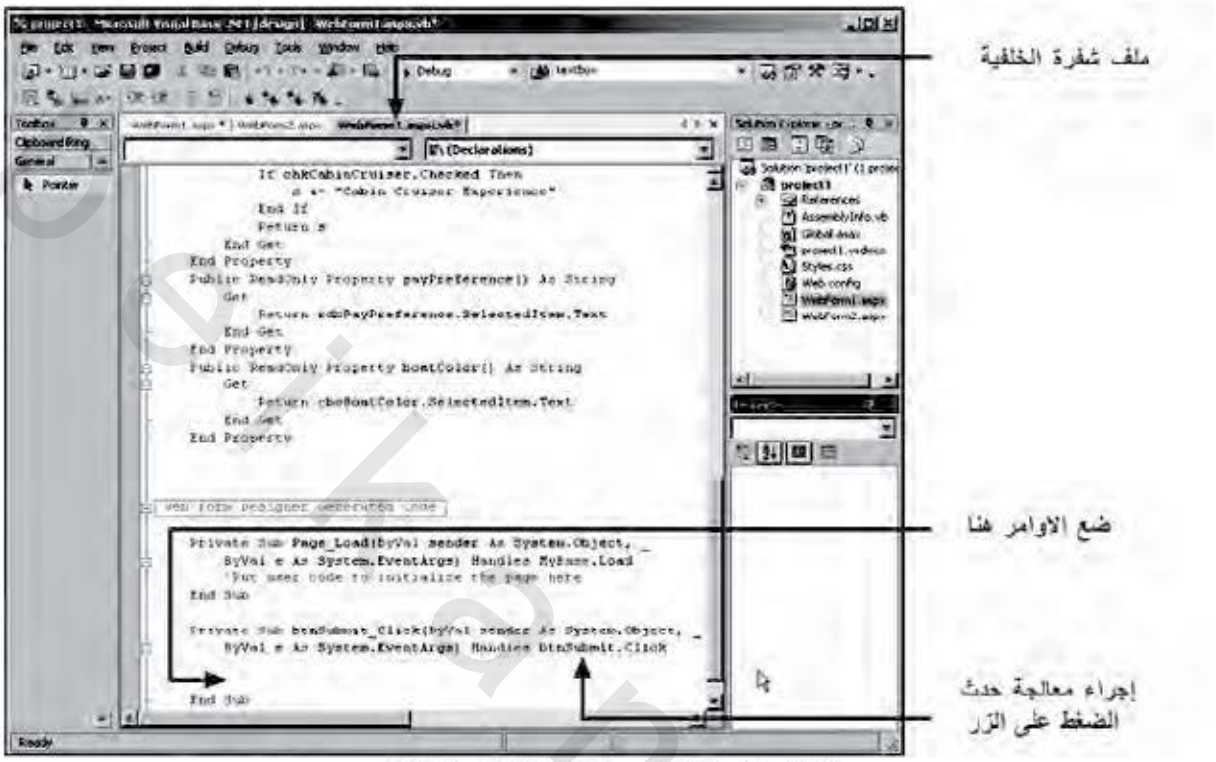
استخدام نافذة الشفرة الخلفية
Using the Code-Behind Window

يوضح هذا الموضوع كيف نكتب أوامر التطبيق الحالي وذلك بإنشاء ملف أخر من النوع aspx.vb الذي سيحتوي على شفرة الخلفية والمسؤولة عن إرسال المعلومات التي تم الحصول عليها من صفحه الويب الأولى إلى صفحه الويب الثانية عند الضغط على الزر Submit، ولفتح نافذة الشفرة الخلفية اضغط ضغطاً مزدوجاً على الزر Submit لفتح النافذة والوقوف على إجراء معالجة حدث الضغط على الزر Submit (الحدث الافتراضي للزر)، ولاحظ أنه يمكن أن نضغط ضغطاً مزدوجاً على أي عنصر آخر لفتح نافذة الشفرة الخلفية والوقوف على إجراء معالجة الحدث الافتراضي لهذا العنصر. يوضح الشكل رقم ١٢,٣٢) ملف نافذة الشفرة الخلفية.

نتجول داخل هذا التطبيق من صفحه الويب الأولى إلى صفحه الويب الثانية حيث يتم إرسال البيانات التي تم الحصول عليها من خلال نموذج الصفحة الأولى إلى الصفحة الثانية، وتذكر أن برتوكول Request/Response يعتمد على أسلوب واحد لعمل ذلك وهو إرسال البيانات (قيم عناصر النموذج داخل المجموعة Form أو المجموعة QueryString المعرفة داخل الكائن Request بناء على الأسلوب المستخدم في إرسال بيانات النموذج (Method="Get" أو Method="Post")، وسنتبع أسلوباً آخر في هذا المثال حيث نعرف مجموعة من الخصائص (Properties) داخل الصفحة الأولى لإرسال قيم عناصر نموذج الصفحة الأولى إلى الصفحة الثانية. أصبح هذا الأسلوب ممكناً الآن لأن صفحة الويب أصبحت صنفاً (الصنف Page)، ومن ثم يمكن إنشاء خصائص داخلها مثل أي صنف آخر.

ولإنجاز هذه المهمة سيتم تعريف مجموعة خصائص للقراءة فقط لعناصر النموذج المراد إرجاع قيمها إلى الصفحة الثانية حيث تعيد كل خاصية قيمة أحد عناصر النموذج، فعلى سبيل المثال توضح الأوامر التالية تعريف خاصية تعيد قيمة صندوق نص الاسم الأول (First Name).

```
Public ReadOnly Property firstName() As String
   Get
      Return txtFirstName.Text
   End Get
End Property
```



الشكل رقم (١٢,٣٢). نافذة شفرة الأوامر الخلفية.

سوف يتم استدعاء هذه الخاصية بواسطة صفحة الويب الثانية لاسترجاع الاسم الأول ومن ثم يمكن تعريف بقية الخصائص بنفس الأسلوب وعندئذ تبقى هناك مهمة واحدة يجب إضافتها داخل الصفحة الأولى وهي استدعاء صفحة الويب الثانية وذلك من خلال استدعاء الإجراء Transfer المعرف داخل الكائن Server كما هو واضح في الأوامر التالية التي ستنفذ استجابة للضغط على الزر Submit في الصفحة الأولى (لاحظ أن اسم الصفحة الثانية هو "Webform2.aspx")

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

   Server.Transfer("Webform2.aspx")
End Sub
```

يوضح الشكل رقم (١٢,٣٣) الشفرة الخلفية لصفحة الويب الأولى حيث توضع السطور الأوامر الناتجة تلقائياً التي تعرف كائنات مع أصناف عناصر النموذج، ثم يظهر تعريف ثلاث خصائص للقراءة فقط الخاصة بصناديق النص الثلاثة الأولى (لاحظ أن التعامل مع عناصر واجهة استخدام الويب من صفات وإجراءات تتشابه

مع التعامل مع عناصر واجهة تطبيقات الويندوز)، ثم تظهر خاصية إرسال قيم صناديق تأشير الخبرة بالمراكب (Boat Experience) في شكل نص (String) وذلك باستخدام أمر If الذي يختبر قيمة صندوق تأشير كل خبرة، ثم يضيف نصاً مناسباً للنص العائد (لاحظ استخدام الرمز
 المسؤول عن إضافة سطر فارغ في النص العائد)، ثم تظهر خاصية إعادة أسلوب الدفع المفضل (Payment Method) حيث استخدمنا الأمر SelectedItem.Text المعرف داخل زر الاختيار، وأخيراً تظهر خاصية إعادة لون المراكب Boat Color حيث استخدمنا الأمر السابق نفسه المعرف داخل قائمة الاختيار المنسدلة.

ثم تأتي بعد ذلك بقية الأوامر الناتجة تلقائياً عند إنشاء صفحه الويب، ولاحظ أنه يمكن غلق تلك المنطقة من الأوامر مثلما نفعل مع بعض نماذج الويندوز، وعلى أي حال يظهر الإجراء InitializeComponent() داخل إجراء معالجه الحدث Page_Init والذي يجب أن يكتب بعده أي أوامر تريد إدراجها، مع العلم أنه لا يوجد احتياج لإضافة أوامر داخل الحدث Page_Init والحدث Page_Load لإضافة أوامر داخل هذا المثال.

وكما ذكرنا سابقاً أن الأمر الوحيد الذي نحتاجه لاستدعاء صفحة الويب الثانية هو استدعاء الإجراء Transfer المعرف داخل الكائن Server، مع العلم أننا لا نحتاج إلى كتابة أوامر لمسح محتويات النموذج عند الضغط على الزر Reset (من النوع HTML الذي يعمل لدى عميل الويب). يوضح الشكل رقم (١٢,٣٤) صفحة الويب الثانية التي ستسترجع البيانات من صفحه الويب الأولى وإظهارها داخل الصفحة.

```
Public Class WebForm1
    Inherits System.Web.UI.Page
    Protected WithEvents Image1 As System.Web.UI.WebControls.Image
    Protected WithEvents txtFirstName As System.Web.UI.WebControls.TextBox
    Protected WithEvents txtLastName As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label3 As System.Web.UI.WebControls.Label
    Protected WithEvents txtDesire As System.Web.UI.WebControls.TextBox
    Protected WithEvents Label4 As System.Web.UI.WebControls.Label
    Protected WithEvents chkSailboat As System.Web.UI.WebControls.CheckBox
    Protected WithEvents chkSkiBoat As System.Web.UI.WebControls.CheckBox
```

الشكل رقم (١٢,٣٣). أوامر الشفرة الخلفية.

```
    Protected WithEvents Label5 As System.Web.UI.WebControls.Label
    Protected WithEvents Label6 As System.Web.UI.WebControls.Label
    Protected WithEvents btnSubmit As System.Web.UI.WebControls.Button
    Protected WithEvents chkCabinCruiser As System.Web.UI.WebControls.CheckBox
    Protected WithEvents rdbPayPreference As
      System.Web.UI.WebControls.RadioButtonList
    Protected WithEvents cboBoatColor As System.Web.UI.WebControls.DropDownList
    Protected WithEvents Label2 As System.Web.UI.WebControls.Label
    Protected WithEvents Label1 As System.Web.UI.WebControls.Label
    Protected WithEvents Image2 As System.Web.UI.WebControls.Image

    Public ReadOnly Property firstName() As String
        Get
            Return txtFirstName.Text
        End Get
    End Property
    Public ReadOnly Property lastName() As String
        Get
            Return txtLastName.Text
        End Get
    End Property
    Public ReadOnly Property slipDesire() As String
        Get
            Return txtDesire.Text
        End Get
    End Property
    Public ReadOnly Property boatExperience() As String
        Get
            Dim s As String
            If chkSailboat.Checked Then
                s &= "Sailboat Experience" & vbCrLf
            End If
            If chkSkiBoat.Checked Then
                s &= "Ski Boat Experience" & vbCrLf
            End If
            If chkCabinCruiser.Checked Then
                s &= "Cabin Cruiser Experience"
            End If
            Return s
        End Get
    End Property
    Public ReadOnly Property payPreference() As String
        Get
            Return rdbPayPreference.SelectedItem.Text
        End Get
    End Property
    Public ReadOnly Property boatColor() As String
        Get
            Return cboBoatColor.SelectedItem.Text
        End Get
    End Property

#Region " Web Form Designer Generated Code "

    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
```

تابع الشكل رقم (١٢,٣٣).

openkandil.com

```
    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()
    End Sub

#End Region

    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
    End Sub

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

        Server.Transfer("Webform2.aspx")
    End Sub
End Class
```

تابع الشكل رقم (١٢,٣٣).

الشكل رقم (١٢,٣٤). صفحة الاستجابة وعرض النتائج.

إنشاء صفحة الويب الثانية لعرض نتائج الاستطلاع
Creating the Results Web Page

نحتاج أن نضيف صفحة ويب ثانية للمشروع الحالي لعرض نتائج الاستطلاع وذلك بالضغط على زر الفأرة الأيمن ثم على اسم المشروع داخل نافذة متصفح الحلول، ثم اختيار الأمر Add ثم اختيار الأمر Add Web Form من القائمة الظاهرة (كما يمكنك اختيار القائمة الفرعية Project ثم الأمر Add ثم Add Web Form). تأخذ الصفحة الجديدة الاسم الافتراضي WebForm2.aspx بشكل تلقائي، والآن نحتاج أن نصمم واجهة النموذج الذي سيظهر نتائج الاستطلاع.

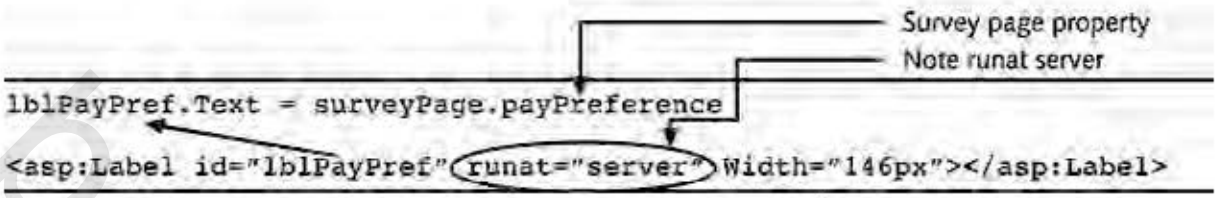
وكما هو واضح من الشكل رقم (١٢,٣٥) فإن تصميم نموذج الصفحة الجديدة يحتوي على جدول لتنسيق عناصر النموذج، كما تتشابه قمة هذا التصميم بقمة تصميم الصفحة الأولى، ويحتوي العمود الأيسر للجدول على عناوين المخرجات التي تم تعريفها من النوع HTML والتي تعمل لدى العميل، وعلى أي حال يظهر العمود الأيمن عناوين من النوع ASP.NET التي تعمل لدى الخادم كما هو واضح من الزر الأيسر العلوي لها، كما يتم تعريف عنصر الجدول من النوع ASP.NET الذي سيعمل لدى الخادم.

الشكل رقم (١٢,٣٥). تصميم صفحة الاستجابة.

توضح الأوامر التالية الشفرة المخفية للنموذج WebForm2 (صفحه الويب الثانية) حيث تم إلغاء الأوامر الناتجة تلقائياً بواسطة مصمم نماذج الويب وذلك بغرض الاختصار، ثم عرفنا متغير الإشارة surveyPage من صنف صفحة الويب الأولى WebForm1، ثم تظهر أوامر إجراء معالجة الحدث Page_Load.

```
Public Class WebForm2
    Inherits System.Web.UI.Page

   'Protected WithEvents Image2 As System.Web.UI.WebControls.Image
   ' Remaining generated code for each control has been deleted
    Dim surveyPage As WebForm1
----Web Form Designer Generated Code goes here---

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

        If Not IsPostBack Then
           surveyPage = CType(context.Handler, WebForm1)

           lblName.Text = surveyPage.firstName & " " & _
               surveyPage.lastName
           lblDesire.Text = surveyPage.slipDesire
           lblBoatExperience.Text = surveyPage.boatExperience
           lblPayPref.Text = surveyPage.payPreference
           lblColor.Text = surveyPage.boatColor
           Table1.BgColor = surveyPage.boatColor

       End If
End Sub
```

تبدأ أوامر هذا الإجراء بالأمر If الذي يختبر الصفة Postpack لتنفيذ جميع أوامر الإجراء عند الاستدعاء الأول لهذه الصفحة فقط، ولم تنفذ في الاستدعاءات التالية لأننا ببساطة لا نريد أن ننفذ الأوامر عند كل استدعاء. يحصل أول أمر (داخل التركيب أمر If) على كائن الصفحة الأولى باستخدام الكائن Context.handler ثم إسناده لمتغير الإشارة surveyPage في بداية أوامر الصفحة ولكن يجب تحويل كائن Context.handler إلى كائن الصفحة الأولى WebForm2، ثم تأتي أوامر استرجاع قيم الخصائص المعرفة في الصفحة الأولى وإسنادها للصفة Text الخاصة بعناصر ASP.NET التي تعمل لدى الخادم والمعرفة في الصفحة الحالية، فعلى سبيل المثال إن العنوان المسؤول عن عرض أسلوب الدفع المفضل لدى العميل هو العنوان lblPayPref، وكما هو واضح من الشكل رقم (١٢,٣٦) قد تم استدعاء الخاصية Pay Preference من الصفحة الأولى وإسنادها للصفة Text الخاصة بهذا العنوان.



الشكل رقم (١٢,٣٦). صفحة الاستجابة وعرض النتائج.

ثم يسند الأمر الأخير لون خلفية الجدول بلون المركب المختار من قبل العميل بالصفحة الأولى وهذا يوضح مدى سهولة التعامل وتغيير عناصر HTML التي تعمل لدى الخادم، وتذكر أن هذا الجدول تم تعريفه من النوع HTML الذي يعمل لدى الخادم حيث تم إضافة الصفة "runat="server إلى شفرة HTML الخاصة به.

ومن هذا المثال يتضح لنا مدى قوة تطوير تطبيقات الويب ومرونتها باستخدام تقنية ASP.NET التي مكنتنا من إدراج عناصر متنوعة سواء كانت تعمل لدى خادم الويب (عناصر ASP.NET وعناصر HTML) داخل نموذج صفحه الويب الواحدة، هذا بالإضافة إلى فصل شفرة الواجهة (شفرة HTML) عن شفرة معالجة الأحداث (الشفرة الخلفية) مما أدى إلى أسلوب برمجة يتشابه مع أسلوب تطوير تطبيقات الويندوز.

استخدام عناصر ASP.NET للتحقق من صحة البيانات
Using ASP.NET Validation Controls

إن مهمة التحقق من صحة البيانات (Data Validation) تمثل مهمة هامة لأي تطبيق يستقبل بيانات من المستخدم حيث يوضح الجدول رقم (١٢,٤) عناصر ASP.NET المسؤولة عن التحقق من صحة البيانات وكيفية استخدامها.

الجدول رقم (١٢,٤). عناصر التحقق من صحة البيانات.

Validation Control	Description
RequiredFieldValidator	Prevents a field from having an entry
CompareValidation	Compares data to a specified value or the value of another control
RangeFieldValidator	Checks to see whether a value is between a lower and upper boundary
RegularExpressionValidator	Checks that the entry matches a pattern defined by a regular expression
CustomValidator	Creates custom code for validation
ValidationSummary	Displays a summary of all the validation controls on the page

يتم التعامل مع عناصر التحقق من صحة البيانات مثلما نفعل مع بقية عناصر ASP.NET وذلك بوضعها داخل النموذج في المكان المناسب ثم ضبط خصائصها باستخدام نافذة الخصائص، فعلى سبيل المثال إذا أردنا استخدام العنصر ReqiuredFieldValidator فيتم وضعه داخل النموذج ثم ضبط الخاصية ControlToValidate بالعنصر المراد التحقق من صحة بياناته، ثم ضبط الخاصية ErrorMessage برسالة الخطأ التي نريد إظهارها عند عدم إدخال قيمة داخل الحقل المراد التحقق من صحة بياناته، مع العلم أن أشهر استخدام لهذا العنصر هو استخدامه مع عناصر صناديق النص للتحقق منه عدم تركها فارغة، كما يمكن أن يستخدم مع قائمة أزرار الاختيار إذا أردنا أن نجبر المستخدم على اختيار أحد هذه الأزرار.

أما العنصر CompareValidation فهو يقارن القيمة المدخلة داخل عنصر ما مثل صندوق النص بقيمة مدخلة في عنصر آخر، فعلى سبيل المثال إذا كان يجب على المستخدم أن يدخل أو يختار تاريخين مختلفين مع العلم أن التاريخ الثاني يجب أن يكون أكبر من التاريخ الأول، فعندئذ يستخدم هذا العنصر للتحقق من ذلك حيث نضبط الخاصية CompareToValidate بالعنصر المراد مقارنة قيمته، ثم نضبط الخاصية ControlToCompare (أو الخاصية ValueToCompare) إذا أردنا المقارنة مع قيمة ثابتة بالعنصر الآخر الذي نريد أن نقارن به (أو بقيمة ثابتة) ثم نضبط الخاصية ErrorMessage برسالة الخطأ، كما يمكننا أن نستخدم نموذجاً ثابتاً من الحروف (Pattern) لكي نتحقق من صيغة القيمة المدخلة مثل صيغة البريد الإلكتروني أو صيغة أرقام الهواتف.

يقدم العنصر RangeFieldValidator كلاً من الخاصية MinimumValue والخاصية MaximumValue لكي نتحقق أن القيمة المدخلة في عنصر ما تقع بين مدى محدد، وأيضاً يقوم العنصر بتقديم الخاصية Type التي تحدد نوع البيانات التي يجب أن تنتمي إليها القيمة المدخلة، ولا ننسى أن نضبط كلاً من الخاصية ControlToValidate والخاصية ErrorMessage كما فعلنا مع العناصر السابقة.

يقدم لك العنصر CustomValidator إمكانية كتابة شفرة أوامر التحقق من صحة البيانات الخاصة بك لعنصر ما حيث يتم تعريف إجراء معالجة الحدث ServerValidate ثم ضبط الخاصية ClientValidationFunction باسم الإجراء الذي يحتوي على أوامر التحقق من صحة البيانات، وبناء على كيفية التحقق من صحة البيانات الخاصة بك يمكن ضبط الخاصية CompareToValidate باسم العنصر الذي تريد التحقق من صحة بياناته أو يتم تركها فارغة.

ولكي نلخص الأخطاء التي تنتج من عدم صحة البيانات المدخلة في عناصر النموذج نستخدم عنصر يسمى ValidationSummary والذي يتم وضعه في أي مكان داخل صفحة الويب لعرض هذه الأخطاء في مكان واحد.

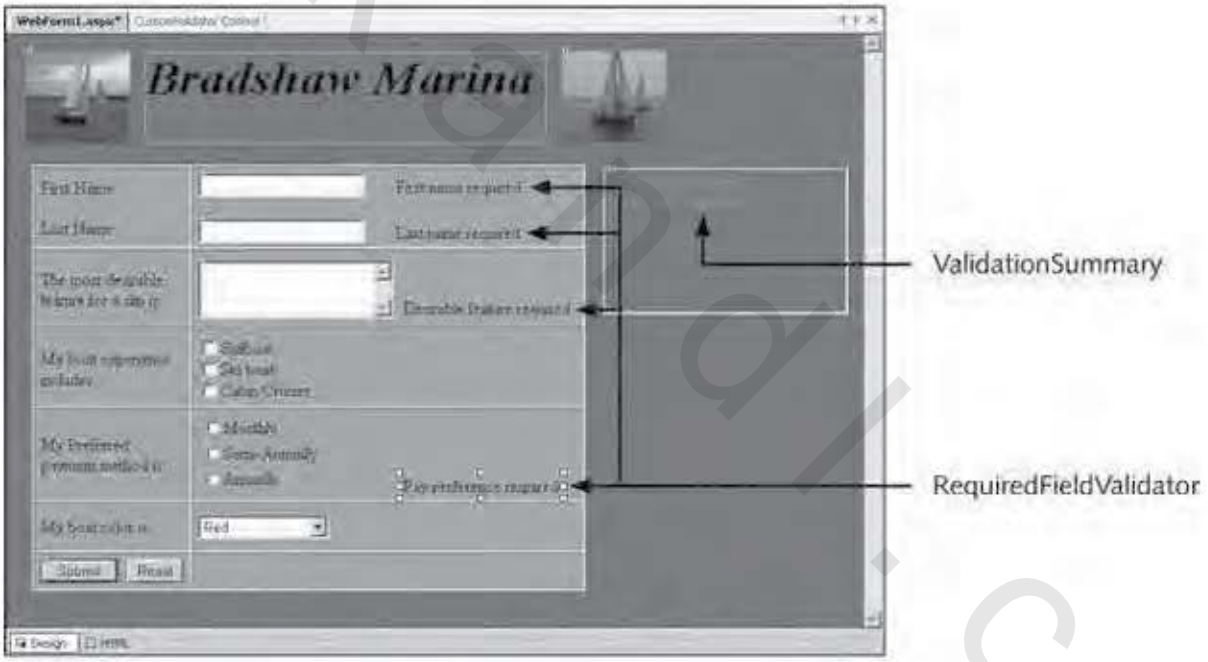
كما يمكنك استخدام أكثر من عنصر للتحقق من صحة بيانات عنصر ما، فعلى سبيل المثال يمكن التحقق من إدخال قيمة داخل عنصر صندوق نص والتحقق أيضاً من أن هذه القيمة تقع بين مدى محدد وذلك باستخدام

كل من العنصر ReqiuredFieldValidator والعنصر RangeFieldValidator، ثم إسناد الخاصية ControlToValidate لكلا العنصرين بعنصر صندوق النص محل الاهتمام.

إضافة عناصر التحقق من صحة البيانات لبرنامج استطلاع الرأي

Adding Validation Controls to the Survey Web Page

يبين المثال التالي أساسيات استخدام عناصر التحقق من صحة البيانات حيث يوضح الشكل رقم (١٢.٣٧) استخدام أربعة عناصر من النوع ReqiuredFieldValidator وعنصر من النوع ValidationSummary الذي تم وضعه داخل عنصر الإطار (Panel) في الركن الأيمن العلوي من النموذج.



الشكل رقم (١٢.٣٧). استخدام عناصر التحقق من صحة البيانات.

تم وضع أول ثلاثة عناصر من النوع RequiredFieldValidator على يمين الصناديق النصية الثلاثة الأولى ("First Name" و "Last Name" و "Desired Slip Features")، وقد تم إضافة العنصر الرابع على يمين عناصر أزرار الاختيار للتأكد أن المستخدم قام باختيار أحدها.

يستخدم عنصر الإطار لاحتواء عنصر تلخيص الأخطاء، ولكن يجب تنسيق عنصر الإطار وحفظه لكي يتلاءم لهذه الوظيفة حيث يجب مسح كلمة "Panel" التي تظهر عند وضعه على صفحة الويب وذلك بالضغط عليها ثم مسحها، ويجب تغيير الصفة BorderStyle الخاصة بهذا العنصر إلى القيمة None.

كما يجب ضبط الخاصية ControlToValidate لعناصر التحقق من صحة البيانات التي تكون من النوع RequiredFieldValidator بالعنصر المناظر لها والمراد التحقق من صحة بياناته (مثل العنصر txtFirstName) حيث يجب أن نضغط على الزر Start لتنفيذ البرنامج بعد الانتهاء من وضع خصائص عناصر التحقق من صحة البيانات وضبطها، فعندئذ يظهر نموذج استطلاع الرأي، فإذا ضغطنا على الزر Submit دون إدخال بيانات ستظهر رسائل الأخطاء كما هو واضح في الشكل رقم (١٢,٣٨).

الشكل رقم (١٢,٣٨). رسائل أخطاء ناتجة من استخدام عناصر التحقق من صحة البيانات.

عزيزي القارئ تذكر أن إجراء معالجة الحدث للزر Submit يحتوي على الأمر Server.Transfer("WebForm2.aspx") المسؤول عن استدعاء صفحة الويب الثانية ؛ ولذلك يمكن الاستفادة من الإجراء isValid المعرف داخل الصنف Page والذي يعيد القيمة True إذا كانت البيانات المدخلة صحيحة والقيمة False إذا كانت البيانات المدخلة غير صحيحة وذلك قبل استدعاء صفحة الويب الثانية كما هو واضح من الأوامر التالية :

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

   If Page.IsValid Then
      Server.Transfer("Webform2.aspx")
   End If

End Sub
```

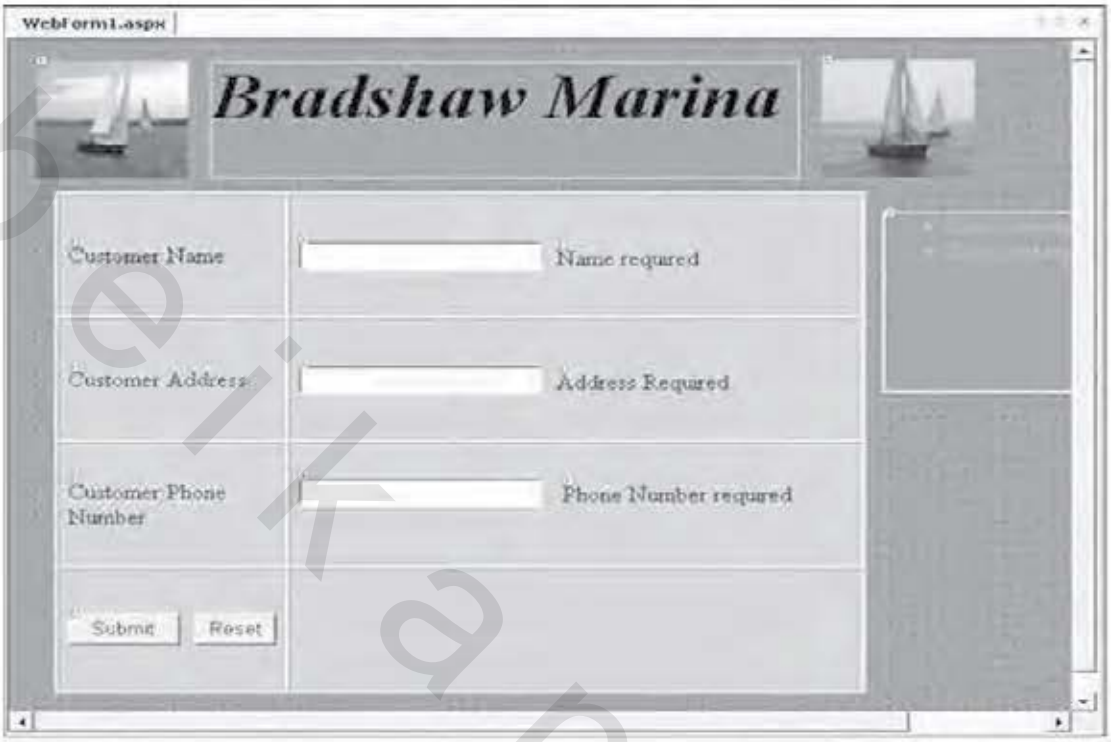
استخدام تقنية ASP.NET مع أصناف مجال المشكلة
Using ASP.NET with a PD Class

يستخدم هذا المثال صنف مجال المشكلة Customer حيث يحتوي هذا التطبيق على نموذج ويب يستخدم لاستقبال بيانات عميل ثم يرسل هذه البيانات إلى صفحة ويب أخرى التي تقوم بإظهارها ثم تعرف كائناً من الصنف Customer ثم تضيفه إلى مصفوفة من النوع ArrayList. يحتوي نموذج الصفحة الثانية على زر عودة إلى الصفحة الأولى لتمكين المستخدم من إضافة عميل جديد.

يحتوي نموذج الويب الثاني على وظيفتين هامتين. تمكّن الوظيفة الأولى المستخدم من عرض بيانات جميع العملاء داخل صندوق نص متعدد السطور وذلك عند الضغط على زر "" ، أما الوظيفة الثانية فتمكّن المستخدم من البحث عن عميل محدد وذلك بإدخال رقم هاتف هذا العميل داخل صندوق النص ثم الضغط على زر Find. يوضح الشكل رقم (١٢,٣٩) نموذج إدخال بيانات عميل جديد، ويوضح الشكل رقم (١٢,٤٠) إظهار بيانات العملاء وذلك بعد إدخال العديد من العملاء.

إن تصميم كل من نموذج إدخال البيانات ونموذج إظهار بيانات العملاء يتشابه مع الأمثلة السابقة عدا أن نموذج إظهار بيانات العملاء يحتوي على صندوق نص متعدد السطور لكي يحتوي على بيانات جميع العملاء حيث يجب تغيير الصفة TextMode لهذا الصندوق إلى القيمة Multiline. سوف يتم استدعاء صفحة بيانات العملاء عند الضغط على الزر Submit في نموذج إدخال البيانات، ومن ثم تظهر بيانات العميل المستقبل داخل العناوين المتواجدة داخل الجدول (لاحظ أن الزر Display All Customers قد تم الضغط عليه لإظهار بيانات جميع العملاء كما هو واضح في الشكل رقم ١٢,٤٠).

ولكي تبحث عن عميل ما، أدخل رقم هاتف هذا العميل داخل صندوق النص ثم اضغط على الزر Find، فعندئذ ستظهر بيانات هذا العميل داخل صندوق النص المتعدد السطور (كما هو واضح في الشكل رقم ١٢,٤١) إذا تم الحصول عليه داخل المصفوفة أو ستظهر رسالة داخل نفس صندوق النص تفيد بأن العميل لم يتم الحصول عليه.



الشكل رقم (١٢,٣٩). نموذج إدخال بيانات عميل.

الشكل رقم (١٢,٤٠). نموذج عرض بيانات العملاء.

obeikandl.com

الشكل رقم (١٢,٤١). نموذج الويب بعد إيجاد عميل.

وكما فعلنا في المثال السابق، يجب أن نعرف مجموعة من الخصائص للقراءة فقط داخل نموذج إدخال البيانات لإمكانية إرسال محتويات صناديق النص إلى صفحة الويب الثانية، كما يجب إضافة أمر استدعاء صفحة الويب الثانية من صفحة الويب الأولى.

توضح الأوامر التالية بداية تعريف صنف صفحة الويب الثانية حيث يتم تعريف متغير إشارة surveyPage لصنف نموذج إدخال البيانات، ثم تعريف المصفوفة Customers من النوع ArrayList لتخزين كائنات من الصنف Customer، ثم تعريف متغيرا إشارة من الصنف Customer (المتغير aCustomer والمتغير cust) حيث يستخدم المتغير cust داخل التكرار For-Each للتجول داخل عناصر المصفوفة Customers، ثم تعريف ثلاثة متغيرات من النوع String لتخزين قيم صفات كائن الصنف Customer.

```
Dim surveyPage As WebForm1

Shared customers As New ArrayList()
```

```
Dim aCustomer As Customer
Dim cust As Customer

Dim name, address, phone As String
```

يستدعي الإجراء Page_Load المعرف داخل نموذج "إظهار بيانات العملاء" الخصائص المعرفة لدى صنف نموذج إدخال البيانات وذلك لاسترجاع بيانات العميل الحالي، ثم عرض هذه القيم داخل عناصر العناوين المعرفة داخل النموذج، ثم تعريف كائن من الصنف Customer باستخدام البيانات المستقبلة لإضافته بعد ذلك داخل المصفوفة Customers كما هو واضح في الأوامر التالية:

```
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
         surveyPage = CType(context.Handler, WebForm1)

         lblName.Text = surveyPage.Name
         lblAddress.Text = surveyPage.Address
         lblPhone.Text = surveyPage.PhoneNo

         aCustomer = New Customer(lblName.Text, _
             lblAddress.Text, lblPhone.Text)

         customers.Add(aCustomer)

     End If
End Sub
```

يستدعي إجراء معالجة حدث الضغط على الزر Return الإجراء Transfer المعرف داخل الصنف Server للعودة إلى نموذج إدخال البيانات مرة ثانية. توضح الأوامر التالية إجراء معالجة الضغط على الزر Display All Customers حيث يستخدم التكرار For-Each للتجول داخل المصفوفة Customers واسترجاع قيم صفات كل عميل باستخدام إجراءات المرور، ثم لصق هذه القيم على قيمة الصفة Text لصندوق النص المتعدد السطور ليظهر كما هو واضح في الشكل رقم (١٢,٤٠).

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

     txtDisplay.Text = "List of Customers" + vbCrLf
     txtDisplay.Text += "==================" + vbCrLf
     For Each cust In customers
```

```
            aCustomer = cust
            ' Could use TellAboutSelf method but couldn't format
            name = aCustomer.GetName
            address = aCustomer.GetAddress
            phone = aCustomer.GetPhoneNo

            txtDisplay.Text += "Name: " & name & vbCrLf
            txtDisplay.Text += "Address: " & address & vbCrLf
            txtDisplay.Text += "Phone No: " & phone & vbCrLf
            txtDisplay.Text += vbCrLf & "==================" & vbCrLf
        Next
End Sub
```

توضح الأوامر التالية إجراء معالجة حدث الضغط على زر Find حيث نستخدم التكرار For-Each للتجول داخل المصفوفة Customers، ثم نختبر قيمة هاتف كل عميل هل تساوي رقم الهاتف المدخل أم لا، فإذا وجدنا عميلاً يتساوى رقم هاتفه مع القيمة المدخلة نقوم بعرض رسالة تفيد عدم الحصول على عميل يملك رقم الهاتف المدخل داخل صندوق النص متعدد السطور.

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

  ' Clear names from labels
  lblName.Text = ""
  lblAddress.Text = ""
  lblPhone.Text = ""

  ' Put values into the text box
  txtDisplay.Text = "Found Customer" + vbCrLf
  txtDisplay.Text += "==================" + vbCrLf

  Dim foundsw As Boolean = False
  txtDisplay.Text = vbNullString

  ' Loop over customers in the ArrayList
  For Each cust In customers
      aCustomer = cust
      phone = cust.GetPhoneNo
      If phone = txtPhone.Text Then
          ' Found customer so display in the textbox
          txtDisplay.Text = "Found Customer" + vbCrLf
          txtDisplay.Text += "==================" + vbCrLf

          name = aCustomer.GetName
          address = aCustomer.GetAddress
          phone = aCustomer.GetPhoneNo

          txtDisplay.Text += "Name: " & name & vbCrLf
          txtDisplay.Text += "Address: " & address & vbCrLf
```

obeikandl.com

```
            txtDisplay.Text += "Phone No: " & phone & vbCrLf
            txtDisplay.Text += vbCrLf & "----------" & vbCrLf
            foundsw = True
        End If
        ' If customer not found display message
        If foundsw = False Then
            txtDisplay.Text = "Customer Not found"
        End If
    Next
End Sub
```

نقل تطبيقات الويب وحل مشاكل الويب

Moving Web Applications and Solving Web Problems

يتطلب نقل تطبيقات الويب إلى إعدادات قبل نشرها على الويب سواء كانت مواقع ثابتة المحتوى أو تطبيقات ASP.NET، ونتعرض داخل هذا الموضوع إلى حل المشكلة التي ربما تحدث عند تشغيل تطبيقات الويب بعد نقلها، مع العلم أننا نفترض أن موقع الويب يتواجد داخل مجلد افتراضي (Virtual Directory) أو مجلد فرعي داخل مجلد افتراضي.

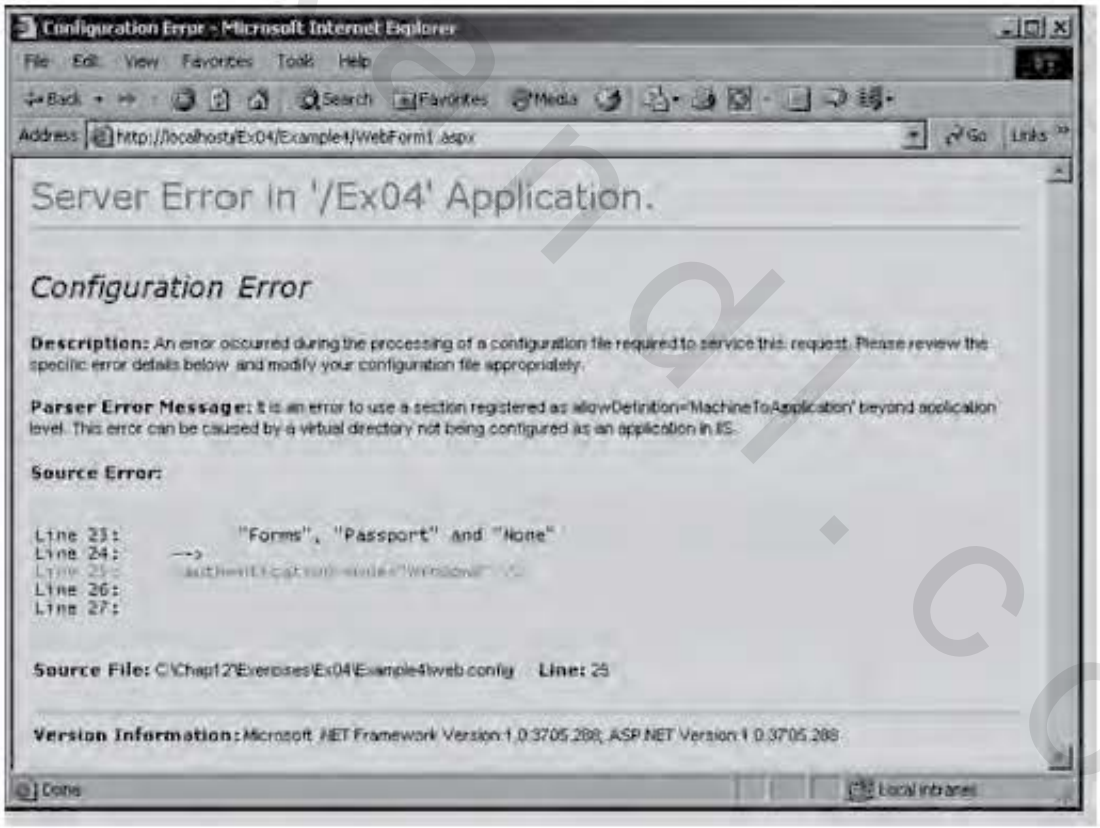
تخزن تقنية ASP.NET عنوان تطبيق الويب داخل ملف XML باسم ممتد ".vbprog.webinfo" (يوجد هذا الملف داخل المجلد الرئيس للمشروع مع باقي الملفات)، فإذا قمت بالضغط المزدوج على هذا الملف ستظهر محتويات الملف كما هو واضح في الشكل رقم (١٢,٤٢). قم بتغيير مسار العنوان الحالي إلى العنوان الجديد، فعلى سبيل المثال افترض أن المشروع الحالي انتقل من المجلد Ex05 إلى المجلد Ex04، إذن قم بهذا التغيير ثم قم بتخزين الملف.

الشكل رقم (١٢,٤٢). مسار عنوان ملف ASP.NET.

وبعد إجراء هذا التغيير اضغط ضغطاً مزدوجاً على الملف ".vbprog" لفتح تطبيق الويب ثم اضغط على زر التشغيل (Start)، وعندما يطلب منك تخزين الملف ".sln" لاستبدال ملف الحل القديم بالملف الجديد وافق على ذلك

ثم اضغط على زر التشغيل مرة ثانية، وفي أغلب الظن ستحتاج إلى أن تعرف صفحة البداية (Start Up Page) ثم شغل البرنامج مرة ثانية، فعندئذ ربما تحصل على رسالة خطأ تشير إلى أن هذا التطبيق لم يتم ضبطه كتطبيق IIS (IIS Application) كما هو واضح في الشكل رقم (١٢,٤٣).

ولضبط التطبيق الخاص بك كتطبيق IIS يجب تشغيل برنامج "Internet Information Services" وذلك بواسطة فتح نافذة لوحة التحكم، ثم فتح نافذة أدوات إدارية، ثم تشغيل البرنامج "Internet Services Manager" (إذا كان نظام تشغيل ويندوز ٢٠٠٠) أو تشغيل البرنامج "Internet Information Services" (إذا كان نظام التشغيل ويندوز XP). اضغط على رمز جهاز الحاسب الآلي الخاص بك في الجزء الأيسر من نافذة البرنامج، ثم اضغط على رمز Web Sites، ثم الرمز Default Web Site (انظر الشكل رقم ١٢,٤٤).



الشكل رقم (١٢,٤٣). خطأ يشير إلى أن التطبيق ليس تطبيق IIS.

الشكل رقم (١٢,٤٤). استخدام نافذة برنامج IIS.

لضبط المثال رقم ٤ كتطبيق IIS، اتبع التالي:

١- ابحث عن المجلد الافتراضي Ex04 (انظر موضوع "إنشاء المجلدات الافتراضية" المذكور سابقاً في هذا الفصل لمعرفة الخطوات التفصيلية لإنشاء مجلدات افتراضية)، ثم اظهر المجلدات الفرعية للفهرس Ex04.

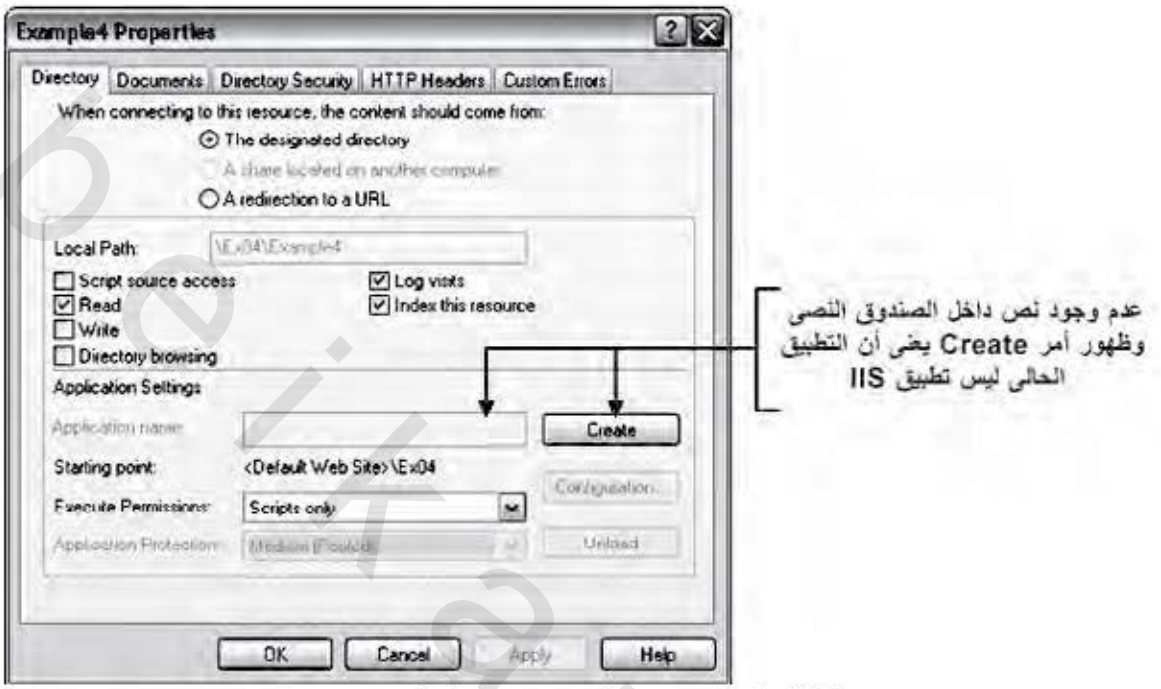
٢- اضغط بزر الفارة الأيمن على المجلد، ثم اختر Properties من القائمة الفرعية لتفتح نافذة بعنوان Properties.

٣- اختر للسطرة Directory كما هو واضح في الشكل رقم (١٢,٤٥).

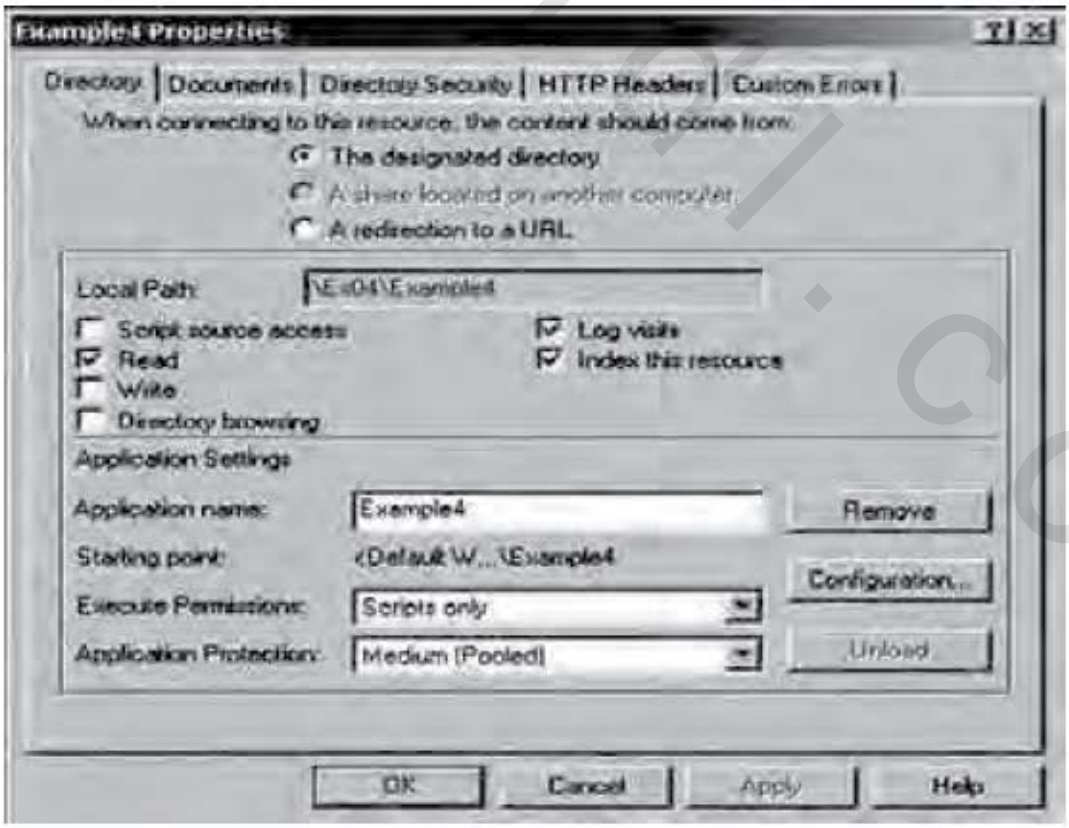
٤- ابحث عن صندوق النص Application والزر الذي على يساره داخل المنطقة "Application Settings"، فإذا لم يكن التطبيق معرفاً كتطبيق IIS ستجد أن صندوق النص فارغ وبأخذ الزر العنوان Create، أما إذا كان التطبيق معرفاً من النوع IIS Application ستجد أن صندوق النص يحتوي على قيمة وبأخذ الزر العنوان Remove، فإذا وجدت صندوق النص فارغاً اضغط على الزر Create، فعندئذ سيتم فتح تعريف المجلد الخاص بك الآن كتطبيق IIS كما هو واضح في الشكل رقم (١٢,٤٦).

٥- اضغط على الزر OK ثم أغلق جميع النوافذ المتعلقة لهذه المهمة. يجب أن يعمل تطبيق IIS الخاص بك الآن بشكل جيد.

وكما رأيت عند نقلك تطبيق ASP.NET من مكان إلى آخر يجب عليك تعريف الموقع الجديد كتطبيق IIS كما فعلنا في الخطوات السابقة.



الشكل رقم (١٢,٤٥). نافذة خصائص المثال رقم ٤.



الشكل رقم (١٢,٤٦). ضبط المشروع كتطبيق من النوع IIS.

ملخص الفصل

Chapter Summary

- تستخدم خدمة الويب على برتوكول Hypertext Transmission Protocol (ويختصر بالاسم HTTP) لمعالجة المعلومات ولغة Hypertext Markup Language (وتختصر بالاسم HTML) لعرض المعلومات على شاشة الحاسب الآلي، ومن ثم يحتاج المستخدم متصفح ويب فقط لكي يستخلص المعلومات ويعرضها على الشاشة حيث تنقسم مواقع الويب إلى مواقع ثابتة المحتوى ومواقع متغيرة المحتوى.
- تعتمد خدمة الويب على نموذج Request/Response المستخدم للتعامل بين كل من عميل الويب وخادم الويب.
- تقدم نماذج لغة HTML على العديد من عناصر إدخال بيانات التي تمكن المستخدم من إدخال البيانات في نموذج، ومن ثم يمكن أن نستخدم المعالجة لدى العميل للتحقق من صحة البيانات قبل إرسالها إلى الخادم.
- تسمح تقنية ASP التقليدية أن تدخل أوامر نصية (Scripts) داخل شفرة HTML التي سوف تنفذ لدى خادم الويب باستخدام عدة لغات نصية (Scripting Languages) مثل لغة VBScript ولغة JavaScript، وتستخدم البرمجة لدى الخادم لاستقبال بيانات من المستخدم ثم استرجاع بيانات من قاعدة البيانات (إذا كانت هناك حاجة لذلك) ثم المعالجة وعودة إرسال نتيجة المعالجة إلى عميل الويب.
- لقد غيرت تقنية ASP.NET أسلوب تطوير تطبيقات الويب وحسنته، إلا أنها مازالت تتوافق مع تقنية ASP، حيث قدمت التقنية عدة مفاهيم أدت إلى أن تطوير تطبيقات الويب باستخدام تقنية ASP.NET يشبه تطوير تطبيقات الويندوز.
- إن أهم ميزة قدمتها تقنية ASP.NET هو فصل شفرة البرنامج ASP.NET عن شفرة محتويات واجهة الصفحة حيث استخدمت تقنية ASP.NET أسلوب Code-Behind الذي يشبه مفهوم البرمجة المعتمدة على الأحداث Event-driven Programming والمتبعة في تطوير تطبيقات الويندوز.
- قدمت تقنية ASP عدة مفاهيم جديدة مثل مفهوم التطوير باستخدام أسلوب "Drag & Drop" وميزة ترجمة شفرة الأوامر بشكل تلقائي (Dynamic Compilation) لتنفيذ أسرع وميزة فصل شفرة البرنامج عن شفرة محتويات وميزة عناصر التحقق من صحة البيانات (Validation Controls).
- يوجد أكثر من أسلوب داخل تقنية ASP.NET لإرسال معلومات من صفحة الويب الأولى إلى صفحة الويب الثانية.

المصطلحات الأساسية

Key Terms

برتوكول Hypertext Transmission Protocol (HTTP)	تقنية ASP التقليدية
لغة Hypertext Markup Language (HTML)	أسلوب Code-Behind
عميل الويب (Web Client)	البرمجة المعتمدة على الأحداث (Event-driven Programming)
خادم الويب (Web Server)	عناصر التحقق من صحة البيانات (Validation Controls)

أسئلة المراجعة

Review Questions

١- HTML هو .....:

(أ) بروتوكول.

(ب) موقع إنترنت.

(ج) لغة العلامات.

(د) خادم الويب.

٢- علامات HTML .....:

(أ) تستخدم لعرض المحتويات.

(ب) لديها علامات تميز.

(ج) لديها علامات تميز فردية.

(د) جميع ما سبق.

٣- متصفح الويب عادة .....:

(أ) يعالج المسافة البيضاء.

(ب) يعالج مسافات الخط الرأسي للمسافة البيضاء.

(ج) يعالج المسافات الأفقية المحتواة في المسافة البيضاء.

(د) يتجاهل المسافة.

٤- علامة HTML الزوجية المستعملة في إدخال البيانات ستكون .....:

(أ) <HEAD></HEAD>.

(ب) <BODY></BODY>.

(ج) <FORM></FORM>.

(د) <TITLE></TITLE>.

obeikandl.com

٥- القيمتان الشائعتان لصفة METHOD للنموذج هما .....:

(أ) SEDND, GET.

(ب) GET, POST.

(ج) SEND, POST.

(د) SUBMIT, RESET.

٦- صفة ACTION لعلامة تميز البداية تستخدم لـ.....:

(أ) تحديد الملف الذي يتم استدعاؤه عند طلب صفحة الويب.

(ب) تحديد طريق الخادم الملائم لإرجاع النتيجة المطلوبة.

(ج) إخبار المستخدم بحالة صفحة الويب التي يستخدمها.

(د) إخبار المستخدم ما هو الفعل الذي يجب عليه أن يأخذه مع البيانات.

٧- غرض زر SUBMIT في تحكم نموذج HTML هو .....:

(أ) لإنعاش المتصفح المستخدم.

(ب) لتأكيد صفحة الويب إلى خادم الويب.

(ج) لاسترجاع البيانات من خادم الويب.

(د) ليكون لديه متصفح يشغل الرسوم المتحركة في كمبيوتر المستخدم.

٨- عند استخدام Request/Response في نموذج الويب، ما هما مجموعتا الطلب المستخدمان في الاحتفاظ بالبيانات التي يستطيع الخادم استخلاصها:

(أ) Form, myCollection.

(ب) QueryString, myCollection.

(ج) Post, GET.

(د) Form, QueryString.

٩- تكنولوجيا مايكروسوفت ASP الكلاسيكية لمعالجة الويب المتغير .....:

(أ) لا تسمح للنص بالاختلاط مع HTML.

(ب) تستطيع استخدام الفيجوال بيسك سكريبت فقط.

(ج) تعمل على جانب-العميل.

(د) تسمح بالاختلاط بالنص و HTML لمعالجة جانب-الخادم.

obeikandl.com

١٠- تطوير ويب الفيجوال بيسك .نت له المعالم الآتية .....:

(أ) يستخدم بيئة تطوير واجهة المستخدم الرسومية drag-and-drop.

(ب) يوفر الانفصال الواضح بين الكود و HTML.

(ج) يحتوي على عناصر التحكم للتحقق من صحة البيانات.

(د) جميع ما سبق.

١١- تستطيع ASP.NET استخدام الآتي ..... في تطوير الويب:

(أ) HTML.

(ب) عناصر تحكم خادم HTML.

(ج) عناصر تحكم خادم ASP.NET.

(د) جميع ما سبق.

١٢- امتداد الملف للرمز-الخفي ASP.NET هو .....:

(أ) aspx.

(ب) aspx.vb.

(ج) config.

(د) css.

١٣- امتداد الملف لصفحة ويب ASP.NET التي تحتفظ بالتقديم والمحتويات هو .....:

(أ) aspx.

(ب) aspx.vb.

(ج) config.

(د) css.

١٤- الطريق الوحيد لمرور القيم من مصدر صفحة ويب ASP.NET إلى هدف صفحة ويب ASP.NET هو .....:

(أ) استخدام إجراء كتابة كائن الإجابة

(ب) إنشاء خصائص لصفحة الهدف وإرسال الصفحة لهذه الخصائص.

(ج) إنشاء خصائص للقراءة-فقط لصفحة المصدر وجعل صفحة الهدف تعمل لهذه الخصائص.

(د) استخدام زر HTML RESET.

obeikandl.com

١٥- ASP.NET لديها عناصر للتحقق من صحة البيانات تسمح لأي نوع من الأنواع الآتية من الاختبارات:

(أ) عدم إدخال المستخدم لأي بيانات.

(ب) تحقق المستخدم من حالة الخادم.

(ج) مدى القيم المدخلة من خلال المستخدم.

(د) "أ" و "ج".

أسئلة المناقشة

Discussion Questions

١- ما هي مميزات استخدام تقنية ASP.NET عن تقنية ASP التقليدية؟

٢- كيف تصمم برنامج ASP.NET الذي يقوم بتجميع معلومات عن العملاء (كائنات الصنف Customer) وتخزينها داخل مصفوفة ArrayList؟

٣- اشرح العبارة "أنه لمن الضروري أن تكون ملماً بلغة HTML عندما تقوم بتطوير تطبيق ASP.NET الذي يعتمد على أسلوب البرمجة المعتمدة على الأحداث (Event-driven Programming)".

٤- صف السيناريوهات المختلفة التي تتبعها عند استخدام كل من عناصر HTML وعناصر خادم ASP.NET؟

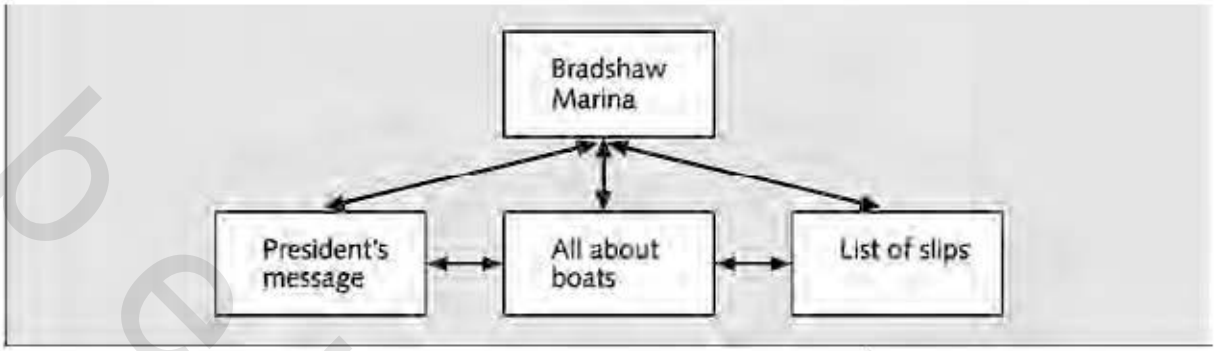
٥- قارن بين كل من تطبيقات الويندوز وتطبيقات الويب ASP.NET. وما هو أفضل نوع تطبيق ويندوز يتناسب مع شركة برادشو مارينا وأفضل نوع تطبيق ويب ASP.NET يتناسب مع الشركة نفسها؟

مشاريع الفصل

Projects

١- قم بتصميم صفحة ويب لشركة برادشو مارينا التي تمكّن المستخدم من تصفح صفحات الويب الموضحة في الشكل رقم (١٢,٤٧). سيشمل هذا المشروع على العديد من الأشياء الهامة التي يجب أن تصممهما مثل صور الأزرار وأربع صفحات ويب (homepage.htm, boats.htm, ListSlips.htm, PresMessage.htm)، ثم قم بتخزين المشروع داخل المجلد Chap12\Projects\Proj01 داخل مجلد العمل الخاص بك.

obeikandl.com



الشكل رقم (١٢,٤٧). التركيب البنائي لموقع شركة برادشو مارينا.

٢- أكمل الخطوات التالية :

- قم بنسخ الملف Bradshaw_Marina_form.htm من المسار "Chap12/Exercises/Ex02/Example2" إلى المجلد "Chap12/Projects/Proj01"، ثم قم بإعادة تسميته إلى test.htm.
- قم بنسخ الملف Bradshaw_Marina_response.asp من المسار "Chap12/Exercises/Ex02/Example2" إلى المجلد "Chap12/Projects/Proj01"، ثم قم بإعادة تسميته إلى queryString.asp.
- قم بتغيير الصفة METHOD إلى GET داخل الملف test.htm.
- قم بتغيير النص "request.form" إلى "request.querystring" داخل الملف queryString.asp.
- قم باختبار الموقع بزيارة الصفحة "http://localhost/*workpath*/test.htm"، ثم قم بكتابة البيانات المناسبة والضغط على زر Submit، ثم قم بملاحظة عنوان الموقع المرسل حيث تم إضافة علامة استفهام متبوعة (?) باسم متغير–قيمة المرسلة ويفصل بينهم بالرمز &.

٣- قم بتطوير برنامج ASP.NET يحتوي على الصنف Boat الذي قمت بتطويره في الفصل التاسع والذي يحتوي على أربع صفات. يجب أن يقدم البرنامج الخدمات التالية :

- يحتوي على نموذج إدخال مركب الذي يستقبل بيانات مركب جديد (الصفات الأربعة) مع إمكانية حذف البيانات المدخلة.
- إضافة إمكانية التحقق من إدخال البيانات.
- إرسال البيانات المدخلة إلى صفحة أخرى.
- إظهار البيانات المرسلة داخل الصفحة الأخرى.
- لاحظ أنه يجب عليك إضافة الصنف Boat داخل هذا البرنامج، كما يجب عليك إزالة أي متغير إشارة لكائنات مجال المشكلة.

٤- قم بإضافة الوظائف التالية للمشروع رقم ٣:

- يجب أن يضيف البرنامج كائنات الصنف Boat إلى مصفوفة من النوع ArrayList.
- يجب أن يحتوي البرنامج على زر لإظهار جميع كائنات الصنف Boat.
- يجب أن يحتوي التطبيق على إمكانية البحث عن مركب بواسطة الصفة "StateRegistrationNo".

٥- قم بإضافة كل من الصنف Dock والصنف Slip إلى المشروع السابق وإزالة أي متغير إشارة لأي كائن مجال مشكلة آخر، ثم قم بتطوير النموذج الذي يمكّنك من إضافة كائنات من الصنف Dock وكائنات من الصنف Slip لكل كائن من الصنف Dock وإضافة نموذج لإظهارها.

obeikandl.com