# الفصل السادس
# أدوات التحقق من صحة المدخلات

*تعرفنا فى الفصلين السابقين على مفهوم أدوات تحكم الويب كما تعرفنا على عدد من الأدوات الهامة. وفى هذا الفصل نتعرف على كيفية التحقق من صحة البيانات التى يقوم المستخدم بإدخالها من خلال هذه الأدوات.*

*بانتهاء هذا الفصل، ستتعرف على:*

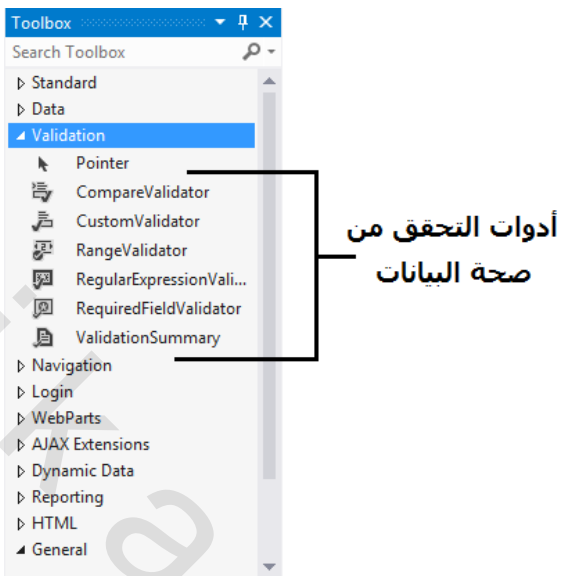
- *استخدام أدوات التحقق من صحة البيانات.*
- *اختبار الحقول المطلوبة.*
- *التحقق على مستوى مدى من البيانات.*
- *مقارنة قيم الحقول.*
- *التحقق من صحة الصفحة بالكامل.*
- *التحكم فى رسائل الخطأ.*
- *التحقق من صحة التعبيرات.*
- *إنشاء دالة تحقق مخصصة.*

بالإضافة إلى استقبال المدخلات من المستخدم، من الضرورى التحقق من صحة هذه المدخلات، حيث يجب أن يحتوى نموذج إدخال البيانات المثالى على كود كافى للتحقق من صحة هذه البيانات. فقبل أن يتم إرسال بيانات المستخدم إلى قاعدة البيانات، يتم أولاً اختبار هذه البيانات وإظهار رسالة الخطأ التى تحث المستخدم على إدخالها مرةً أخرى إذا لم تكن صحيحة.

## أدوات التحقق من صحة البيانات

لا تحتاج بعض أدوات التحكم إلى المزيد من التحقق من صحة بياناتها لأنها تكون أساساً مقيدة بمجموعة معينة من البيانات. فمثلاً، تحتوى أداة مربع السرد دائماً على مجموعة من الخيارات التى يقوم المستخدم بالاختيار من بينها وبالتالى لا يوجد مجال لعملية الخطأ. أما مربع النص **TextBox** فهو على العكس من ذلك يتيح للمستخدم إدخال ما شاء من البيانات الحرفية والرقمية، لذا فمن المنطقى التأكد من صحة بياناته قبل نقلها إلى قاعدة البيانات الموجودة على الخادم. ولمساعدتك فى التحقق من صحة بيانات المستخدم على الويب، أتاح لك نطاق **.NET** مجموعة من التصنيفات التى يمكنك استخدامها فى نماذج الويب فى صورة عدد من أدوات التحكم التى يمكن إجمالها فيمايلى (انظر شكل ٦-١):

- *الأداة* ***RequiredFieldValidator*** والتى تقوم باختبار إدخال المستخدم لأى قيمة داخل أحد الحقول.
- *الأداة* ***CompareValidator*** والتى تقوم بالمقارنة بين حقلين أو مقارنة أحد الحقول مع قيمة معينة.
- *الأداة* ***RangeValidator*** وتقوم بمعرفة مدى وقوع إحدى القيم داخل مدى معرف من القيم.
- *الأداة* ***RegularExpressionValidator*** وتقوم بالتأكد من توافق قيمة أحد الحقول مع تعبير منتظم.



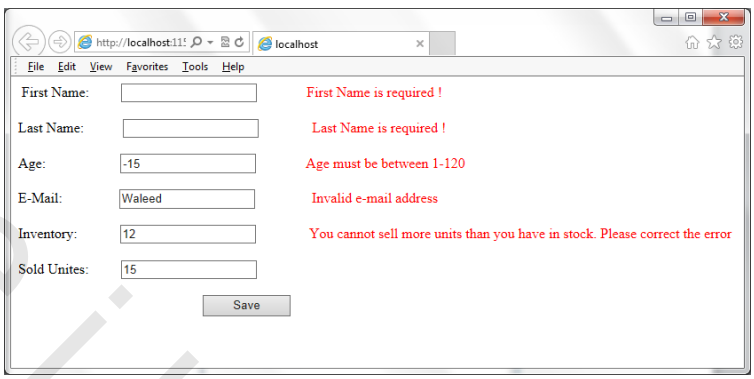

شكل ٦-١ عناصر تحكم التحقق من صحة البيانات

- *الأداة* ***CustomValidator*** وتتيح لك إضافة كود مخصص لتحديد مدى صحة أحد حقول الإدخال.
- *الأداة* ***ValidationSummery*** وتقوم بجمع رسائل الخطأ من عدد من أدوات تحكم التحقق من الصحة.

ولعل الهدف الأساسى من جميع هذه الأدوات هو تبسيط عملية اختبار البيانات التى يقوم المستخدم بإدخالها. وعند تصميم صفحة الويب، تظهر هذه الأدوات كأدوات عنوان تحتوى على الرسائل المناسبة لعدم التحقق من الصحة.

## استخدام أدوات التحكم من صحة البيانات

لتوضيح مدى سهولة استخدام أدوات التحقق من الصحة السابقة داخل تطبيقات الويب، سنقوم بإنشاء مثال بسيط لصفحة ويب تحتوى على مجموعة من مربعات النصوص (انظر شكل ٦-٢).




شكل ٦-٢ نموذج الويب المزمع إنشاؤه

تابع معنا الخطوات الآتية:

١. قم بإنشاء تطبيق ويب جديد باسم مناسب وليكن Validationمع اختيار .NET Framework 3.5 بدلاً من .NET Framework 4.5 .

٢. قم بإضافة نموذج ويب جديد إلى التطبيق كما سبق باسم Default.

٣. قم بإضافة ستة مربعات نصوص إلى النموذج بالأسماء الآتية: txtFirstName وtxtLastName وtxtAge وtxtEMail وtxtInventory وtxtSoldUnits.

٤. قم بحذف النص الافتراضى الموجود فى مربعات النص إن وجد حتى لا تظهر أى نصوص داخل هذه المربعات عند فتح النموذج داخل المستعرض.

٥. قم بوضع أداة عنوان يسار كل مربع من مربعات النصوص الستة لشرح محتوياته، كما يمكنك أيضاً كتابة العناوين مباشرةً قبل كل مربع.

٦. قم بوضع زر أمر Button بالجزء السفلى من النموذج مع تخصيص القيمة Save للخاصية Text والقيمة btnSave للخاصية ID.

وبذلك قد أضفنا الحقول الأساسية للنموذج. الخطوة التالية هى إضافة أداة تحقق من الصحة لكل حقل من هذه الحقول حيث تختلف الأداة المستخدمة باختلاف نوع العملية والهدف منها.

## اختبار الحقول المطلوبة

ربما صادفتك العديد من الأمثلة على الحقول المطلوبة أثناء عملك فى الحياة العملية. فلا يمكن مثلاً أن تقوم بسحب مبلغ من المال من حسابك بالبنك دون أن تحدد كمية هذا المبلغ. كذلك الحال فى المثال الذى بين أيدينا، من الضرورى عدم ترك مربعات النصوص التى تعبر عن الاسم الأول والاسم الأخير للمستخدم خالية. للتحقق من صحة هذه البيانات وإظهار الرسالة المناسبة للمستخدم فى حالة خطئها، تابع معنا الخطوات الآتية:

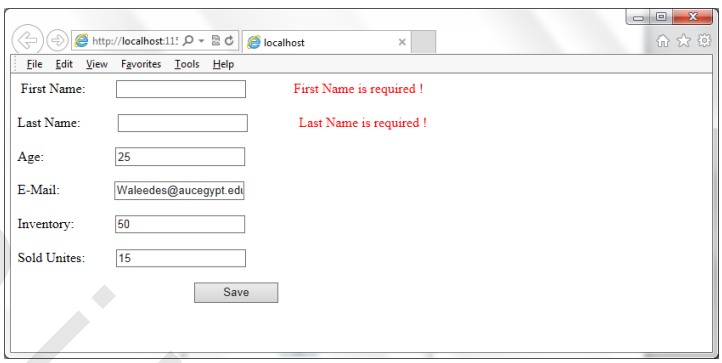
١. من مربع الأدوات، قم بإضافة أداتى تحكم **RequiredFieldValidator** إلى النموذج، إحداهما يمين مربع النص **First Name** والأخرى يمين مربع النص **Last Name** (راجع شكل ٦-٢).

٢. من نافذة الخصائص المصاحبة لكل أداة من أداتى التحقق من الصحة،قم بتخصيص القيمة **valFirstName** للخاصية **ID** لأداة التحكم الأولى والقيمة **valLastName** لأداة التحكم الثانية.

٣. قم بتخصيص القيمة **First Name is required !** للخاصية **ErrorMessage** للأداة الأولى والقيمة **Last Name is required !** للأداة الثانية.

٤. قم باختيار أداة التحكم المصاحبة لكلٍ من أداتى التحقق من الصحة التى قمت بإضافتها وذلك من خلال الخاصية **ControltoValidate** التى تحتوى على مربع سرد يمكنك من خلاله اختيار الأداة المصاحبة.

قم الآن بتشغيل البرنامج. وحينما يظهر النموذج داخل مستعرض الويب، اضغط زر **Save** دون أن تقوم بإدخال أية بيانات بمربعى النص **First Name** و**Last Name**. ماذا تلاحظ ؟ يتم إظهار رسائل الخطأ التى قمت بتحديدها بجوار مربع النص بمجرد تحديث الصفحة (انظر شكل ٦-٣).




شكل ٦–٣ تظهر رسائل الخطأ لعدم إدخال بيانات فى مربعى النص

## التحقق على مستوى مدى من البيانات

نريد أحياناً أن نتأكد من وقوع القيمة المدخلة من المستخدم ضمن مدى معين من القيم. وحقل العمر Age فى النموذج الذى بين أيدينا على سبيل المثال خير مثال على ذلك. فمن الطبيعى أن تقع جميع أعمار المستخدمين بين ١ و ١٢٠ ومن غير الطبيعى إدخال قيمة سالبة أو قيمة رقمية مثل العدد 2,000. لأداء ذلك، يمكنك استخدام الأداةRangeValidator وتخصيص إعداداتها المختلفة بنفس طريقة الأداة السابقة، الفرق الوحيد هو إضافة بعض الخصائص التى تتحكم فى مدى القيم المحدد. لإجراء التحقق من الصحة على مدى معين من البيانات (عمر المستخدم) فى المثال الذى بين أيدينا، تابع معنا الخطوات الآتية:

١. قم بإدراج الأداة RangeValidator من مربع الأدوات إلى النموذج يمين مربع النص Age.
٢. قم بتغيير اسم الأداة من خلال الخاصية ID إلى اسم مناسب وليكن valAge.
٣. انقر النموذج نقراً مزدوجاً ثم قم بإدخال الكود التالى داخل إجراء الحدث Page_Load:

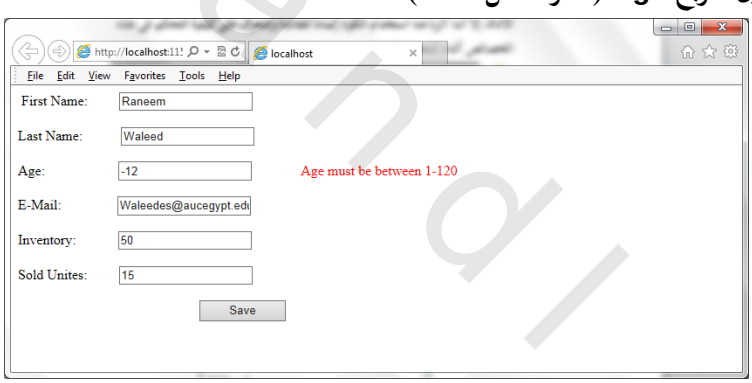
```
valAge.ControlToValidate = "txtAge"
valAge.ErrorMessage = "Age must be between 1-120"
valAge.Type = ValidationDataType.Integer
valAge.MinimumValue = 1
```

```
valAge.MaximumValue = 120
```

وعلى الرغم من إمكانية استخدام مربع الخصائص فى تعيين الخصائص المختلفة لهذه الأداة، إلا أننا آثرنا هنا استخدام الكود إتماماً للفائدة وللتعرف على كيفية التحكم فى هذه الخصائص أثناء تشغيل التطبيق.

من الضرورى تحديد نوع البيانات التى يتم اختبارها بواسطة الأداة RangeValidator وذلك لأن معالجة الأرقام الصحيحة مثلاً تختلف كثيراً عن معالجة الحروف وسلاسل البيانات. فالحرف 2 مثلاً لا يقع فى المدى من ١ إلى ١٢٠، لذا تحرى دائماً اختيار نوع البيانات المناسب.

قم الآن بتشغيل التطبيق مع إدخال عمر لا يقع فى المدى ١-١٢٠، تلاحظ ظهور رسالة الخطأ يمين المربع Age (انظر شكل ٦-٤).




شكل ٦-٤ إدخال رقم لا يقع فى المدى ١-١٢٠

## مقارنة قيم الحقول

تعتبر الأداة **CompareValidator** إحدى الأدوات الهامة للتحقق من الصحة حيث تستخدم لإجراء مقارنة حسابية بين القيمة الموجودة بمربع النص وقيمة مقارنة أخرى كما يمكن استخدامها أيضاً للتحقق من صحة نوع بيانات الحقل.

ولا تختلف خطوات إضافة هذه الأداة إلى النموذج وتعيين خصائصها كثيراً عن الأدوات السابقة، وإنما الجديد هنا هو تحديد نوع المقارنة من خلال الخصائص **Operator** و**Type** ثم تحديد قيمة المقارنة تبعاً للقواعد العامة التالية:

- عند إجراء المقارنة بين قيمتين فى مربعى نصوص، قم بتخصيص اسم المربع الثانى للخاصية **ControlToCompare** ولا تقم بتعيين الخاصية **ValueToCompare**.
- عند إجراء مقارنة بين محتويات الحقل وقيمة أخرى ثابتة، قم بتخصيص هذه القيمة للخاصية **ValueToCompare** ولا تقم بتعيين الخاصية **ControlToCompare**.
- عند التأكد من صحة بيانات مربع النص، فلا تقم بتعيين أيٍ من الخاصيتين **ValueToCompare** و**ControlToCompare** ولكن قم بتعيين نوع البيانات داخل الخاصية **Type**.

يمكنك حقيقةً تصور عملية المقارنة كما لو كانت عبارة **If** مع وجود قيمة الخاصية **ControlToValidate** باليسار ثم معامل المقارنة ثم قيمة المقارنة.فإذا كانت النتيجة **True**، تكون العملية ناجحة وإلا يتم إظهار رسالة الخطأ المحددة كما يلى:

```
If ControlToValidateopearator value Then
        continue
Else
        ErrorMessage
End If
```

سنقوم فيما يلى بالتأكد من أن القيمة المباعة أقل من أو تساوى القيمة الموجودة بالمخزن.

تابع معنا الخطوات الآتية:

١. قم بإضافة الأداة **CompareValidator** من مربع الأدوات إلى النموذج يمين مربعى النص **Inventory** و**Sold Units**.

٢. أعد تسمية الأداة إلى **valCompare** من خلال الخاصية **ID**.

٣. قم بتخصيص عبارة الخطأ التالية للخاصية **ErrorMessage**:

**You cannot sell more units than you have in stock. Please correct the error**

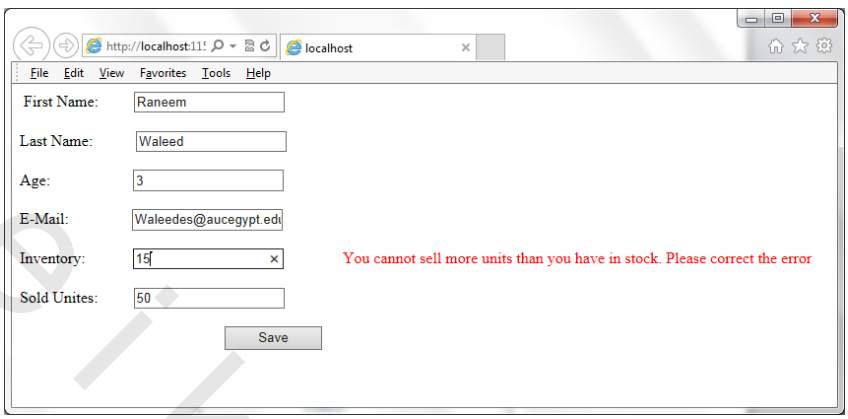
٤. قم بتخصيص مربع النص الثانى **txtSoldUnits**إلى الخاصية **ControlToValidate**.

٥. قم بتخصيص مربع النص الأول **txtInventory**إلى الخاصية **ControlToCompare**.

٦. قم باختيار القيمة **LessThanEqual** داخل الخاصية **Operator**.

تعتبر الأداة **CompareValidator** أن القيمة الخالية لمربع النص صحيحة. فإذا أردت التأكد من وجود قيمة داخل المربع، يمكنك إضافة الأداة **RequiredFieldValidator** واستخدامها مع نفس المربع.

والآن قم بتشغيل التطبيق مع إدخال قيمة داخل مربع النص **Sold Units** أقل من القيمة الموجودة بمربع النص **Inventory**، تلاحظ ظهور رسالة الخطأ التى قمنا بإدخالها فى الخطوات السابقة (انظر شكل ٦-٥).




شكل ٦-٥ يجب أن تكون الكمية المباعة أقل من أو تساوى الكمية الموجودة بالمخزن

## التحقق من صحة الصفحة بالكامل

لعلك الآن تعلم علم اليقين أن استخدام أدوات تحكم التحقق من الصحة يمكنك من إشعار المستخدم بالبيانات الخاطئة التى يقوم بإدخالها. ولكن يبقى القرار النهائى لإرسال البيانات إلى قاعدة البيانات من عدمه فى يدك من خلال الكود الذى تقوم بكتابته.

تحتوى كل أداة من أدوات التحقق من الصحة على الخاصية **IsValid** والتى تكون قيمتها **True** إذا نجحت عملية الاختبار بينما تكون هذه القيمة **False** فى حالة العكس لا قدر الله. يمكنك استخدام الخاصية **IsValid** الخاصة بالكائن **Page** الذى يعبر عن الصفحة بالكامل لاختبار جميع أدوات التحقق من الصحة الموجودة بالصفحة وذلك كما فى الكود التالى الخاص بحدث نقر الزر **Save**:

```
Private Sub btnSave_Click(sender As Object, e As EventArgs)
Handles btnSave.Click
If Page.IsValid Then
  Call YourSaveDataFunction()
Response.Redirect("http://www.compuscience.com.eg")
End If
End Sub
```

يتم فى هذا الكود استدعاء دالة مخصصة باسم **YourSaveDataFunction()** لتخزين بيانات المستخدم ثم الذهاب إلى صفحة أخرى، هذا فقط فى حالة صحة هذه البيانات وذلك

من خلال الخاصية **IsValid**.

## التحكم فى رسائل الخطأ

لعلك لاحظت فى المثال الذى بين أيدينا أننا نقوم بوضع أدوات التحقق من الصحة يمين الحقول التى يتم التحقق من صحتها وبالتالى تظهر الرسائل دائماً يمين هذه الحقول وهذا هو النمط النموذجى لصفحات الويب، وهذا فى حالة وجود متسع داخل الصفحة لعرض هذه الرسائل. أما فى حالة عدم توفر هذه المساحة، فيمكنك استخدام الأداة **ValidationSummery**والتى لا ترتبط بحقل معين داخل الصفحة وإنما تقوم بعرض رسائل الخطأ الموجودة بأدوات التحقق من الصحة الموجودة بالصفحة إذا كانت قيمة الخاصية **IsValid** بها **False** وهو ما يعنى خطأ البيانات التى قام المستخدم بإدخالها.

ولأن نص أداة التحكم **ValidationSummery** مبنى أساساً على رسائل الخطأ الموجودة بالأدوات الأخرى، فمن الضرورى تعيين هذه الرسائل بتلك الأدوات إلى جانب إمكانية إضافة عنوان مخصص للأداة باستخدام الخاصية **HeaderText** وتحديد نمط عرض النص المختصر من خلال الخاصية **DisplayMode** كما فى الكود التالى:

```
valSum.HeaderText = "Your request cannot be processed
because: "
valSum.DisplayMode = ValidationSummeryDisplayMode.
BulletList
```

تدعم بعض مستعرضات الويب إظهار نص الأداة **ValidationSummery** داخل مربع رسالة وذلك بتخصيص القيمة **True** للخاصية **ShowMessageBox** كما يمكنك تعطيل هذه الأداة وعدم إظهار محتوياتها داخل المستعرض عن طريق تخصيص القيمة **False** للخاصية **ShowSummery** الخاصة بالأداة.

## التحقق من صحة التعبيرات

تقوم معظم أدوات التحقق من الصحة الموجودة داخل نطاق **ASP.NET** بأداء العمليات البسيطة، إلا أن النموذج البرمجى فى جعبته العديد من الاختبارات المركبة من خلال إمكانية التحقق من صحة التعبيرات المعتادة**Regular Expressions** والتى تأخذ تنسيقات معينة، هذا بالإضافة إلى إمكانية برمجة الكود الخاص بك لأداء المزيد من المهام.

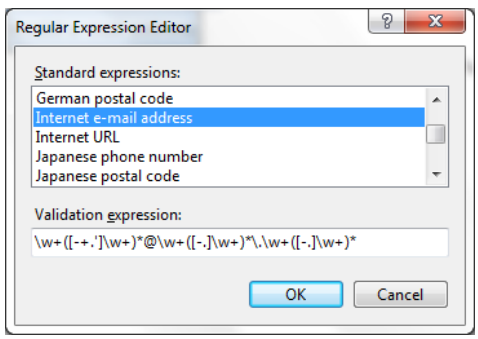
يمكنك استخدام الأداة **RegularExpressionValidator** لمقارنة قيمة مربع النص بأحد التعبيرات المعتادة. ففى المثال الذى بين أيدينا، لدينا حقل لإدخال البريد الإلكترونى الخاص بالمستخدم. وكما نعرف فإن البريد الإلكترونى يحتوى دائماً على الاسم متبوعاً بالعلامة @ ثم اسم مجال الإنترنت المستخدم هكذا مثلاً:

**Waleedes@Aucegypt.edu**

*استخدام التعبيرات المعتادة*

سنقوم فيما يلى بالتعرف على كيفية استخدام هذه الأداة للتحقق من صحة البريد الإلكترونى الذى يقوم المستخدم بإدخاله. تابع معنا الخطوات الآتية:

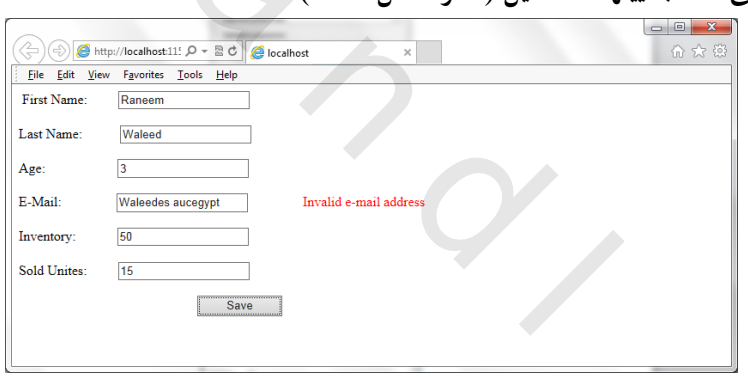
١. قم بإضافة الأداة **RegularExpressionValidator** من مربع الأدوات إلى النموذج يمين مربع النص **Email**.
٢. قم بتغيير اسم الأداة إلى اسم معبر وليكن **valRegular** من خلال الخاصية **ID** المصاحبة للأداة.
٣. قم بتخصيص القيمة **txtEmail** للخاصية **ControlToValidate** المصاحبة للأداة.
٤. قم بتخصيص العبارة **Invalid e-mail address** للخاصية **ErrorMessage**.
٥. انقر الزر المجاور للخاصية **ValidationExpression**، يظهر المربع الحوارى **Regular Expression Editor** (انظر شكل ٦-٦).




شكل ٦-٦ اختيار التعبير المناسب

٦. اختر **Internet e-mail address** من قائمة التعبيرات القياسية ثم انقر زر **Ok**.

والآن قم بتشغيل التطبيق مع إدخال بريد إليكترونى غير صحيح، وفى هذه الحالة تظهر رسالة الخطأ التى قمنا بتعيينها منذ قليل (انظر شكل ٦-٧).




شكل ٦-٧ تظهر رسالة الخطأ فى حالة إدخال بريد إليكترونى غير صحيح

## إنشاء دالة تحقق مخصصة

إذا لم تلبى أى من أدوات التحقق من الصحة الموجودة داخل **ASP.NET** متطلباتك، يمكنك إنشاء دالة مخصصة للتحقق من الصحة من خلال الأداة **CustomValidator** والتى تقوم تلقائياً باستدعاء هذه الدالة عند الضرورة والتى تقوم بدورها هى الأخرى بتخصيص القيمة المناسبة للخاصية **IsValid**.

يمكنك حقيقةً كتابة كل ما تريد من وظائف داخل الدالة المخصصة بما فى ذلك البحث فى

قاعدة البيانات. يوضح الكود التالى دالة مخصصة تقوم بالتأكد من أن القيمة المدخلة فى مربع النص عبارة عن يوم من أيام الأسبوع.

```
Private Sub CheckWeekDay(source As Object, args As
ServerValidateEventArgs)
  Dim UserInputAs String = args.Value
  If IsDate(UserInput) Then
     Dim thedate As Date = Convert.ToDateTime(UserInput)
     If thedate.DayOfWeek = DayOfWeek.Saturday Or
thedate.DayOfWeek = DayOfWeek.Sunday Then
args.IsValid = False
     Else
args.IsValid = True
     End If
  Else
args.IsValid = False
  End If
  End Sub
```

يحتوى المعامل الثانى الممرر للدالة المخصصة على خاصيتين هامتين، الأولى هى الخاصية **Value** والتى تتضمن القيمة التى قام المستخدم بإدخالها، والثانية عبارة عن الخاصية **IsValid** والتى يقوم كود الدالة المخصصة بتعيينها بناءً على مدى صحة البيانات من عدمه. وكى تعمل هذه الدالة بشكل صحيح، يجب إضافة معالج للحدث الموجود على الخادم كما يلى:

```
AddHandlervalMeetingDate.ServerValidate,
AddressOfCheckWeekDay
```

حيث **valMeetingDate** عبارة عن اسم الأداة **CustomValidator**.