

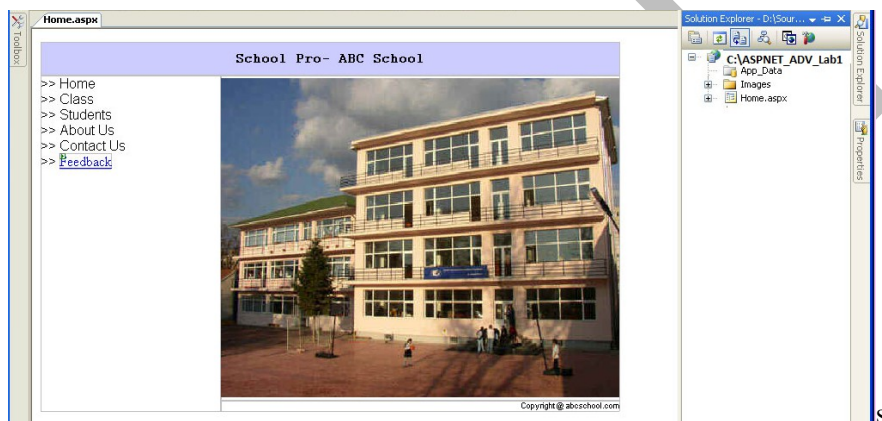
BÀI THỰC HÀNH LẬP TRÌNH ASP.NET

Bài thực hành lập trình ASP.NET nâng cao – Bài 1

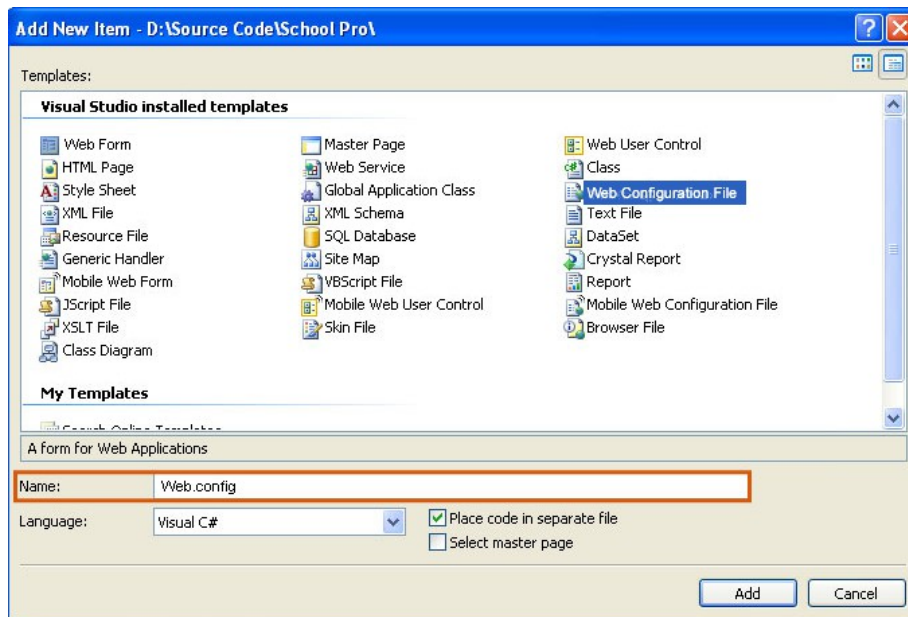
Trong bài thực hành này bạn sẽ thực hiện các thao tác cấu hình cho Ứng dụng Web theo các yêu cầu dưới đây:

- Thêm mới một file **Web.config** vào dự án
- Khai báo chuỗi kết nối
- Thiết đặt chế độ Authentication sử dụng công cụ Web Site Administration
- Thiết đặt thông tin trạng thái phiên làm việc bao gồm sử dụng cookies, thời gian thiết đặt cho phiên làm việc là 20 phút và chế độ (session mode là **InProc**)
- Chuyển tiếp người dùng chưa được chứng thực tới trang Login
- Hiển thị trang thông tin thông báo lỗi thay vì trang thông tin lỗi chung.

1. Mở Ứng dụng Web từ thư mục ASPNET_ADV_Lab1 chỉ định



2. Trong **Solution Explorer** thêm mới một file Web.config vào trong Ứng dụng Web.



3. Trong file **Web.config**, xóa toàn bộ nội dung và gõ thông tin cấu hình như sau

```

1 <?xml version="1.0"?>
2
3 <configuration>
4   <connectionStrings>
5     <add name="sproConn" connectionString="Data Source=.;Database=SMS;providerName=System.Data.SqlClient"/>
6   </connectionStrings>
7 </configuration>
8

```

Lưu ý: chuỗi kết nối có phải sửa theo thông tin cấu hình của từng máy thực hành.

4. Thêm mới một Web Form vào trong dự án và đặt tên là **TestConnection.aspx**
5. Trong **Solution Explorer** bấm chuột phải lên **TestConnection.aspx** và chọn **View Code**.
6. Trong sự kiện **Page_Load** nhập mã lệnh để nhận thông tin kết nối

```

protected void Page_Load(object sender, EventArgs e)
{
    try
    {
        String constring = WebConfigurationManager.ConnectionStrings["sproConn"].ToString();
        SqlConnection con = new SqlConnection(constring);
        Response.Write("Connection Established Successfully");
    }
    catch (SqlException ex)
    {
        Response.Write("Connection Failed !!!");
    }
}

```

7. Trong **Solution Explorer** Bấm đúp vào File **Web.config** để mở chế độ code
8. Khai báo thông tin cấu hình Session State như hình dưới đây.

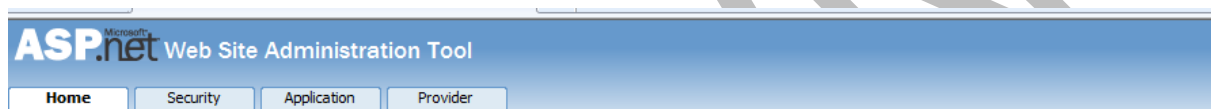
```

1 <?xml version="1.0"?>
2
3 <configuration>
4   <connectionStrings>
5     <add name="sproConn" connectionString="Data Source=.;Database=SMS;providerName=System.Data.SqlClient"/>
6   </connectionStrings>
7   <system.web>
8     <sessionState mode="InProc" cookieless="false" timeout="20">
9   </sessionState>
10  </system.web>
11 </configuration>
12

```

9. Trong **Solution Explorer** bấm chuột phải lên file **TestConnection.aspx** và chọn **View in Browser**. Và kiểm tra kết quả.

10. Trong cửa sổ Visual Studio, bấm vào **Website** menu và chọn **ASPNET Configuration** một trang cấu hình xuất hiện tương tự như hình dưới đây:



Welcome to the Web Site Administration Tool

Application: /ASPNET_ADV_Lab1

Current User Name: TSB\DAO QUANG DUNG

| | |
|---|--|
| Security | Enables you to set up and edit users, roles, and access permissions for your site. Site is using windows authentication for user management. |
| Application Configuration | Enables you to manage your application's configuration settings. |
| Provider Configuration | Enables you to specify where and how to store administration data used by your Web site. |

11. Bấm chọn Tab **Security** như hình dưới đây

You can use the Web Site Administration Tool to manage all the security settings for your application. You can set up users and passwords (authentication), create roles (groups of users), and create permissions (rules for controlling access to parts of your application).

By default, user information is stored in a Microsoft SQL Server Express database in the Data folder of your Web site. If you want to store user information in a different database, use the Provider tab to select a different provider.

[Use the security Setup Wizard to configure security step by step.](#)

Click the links in the table to manage the settings for your application.

| Users | Roles | Access Rules |
|--|---|--|
| Existing users: 0 Create user Manage users Select authentication type | Roles are not enabled Enable roles Create or Manage roles | Create access rules Manage access rules |

12. Bấm chọn **Select authentication type** để thay đổi phương thức Authentication thành Form the internet và chọn nút **Done**



How will users access your site?

☐ From the internet

Select this option if users will access your web site from the public internet. Users will be required to log on using a web form. The site will use forms authentication to identify users according to user information that you store in a database.

☒ From a local network

Select this option if users will access your web site only from a private local network. The site will use built-in Microsoft Windows authentication to identify users. Users with a valid Windows user name and password will be able to access your site.

13. Bạn hãy mở file Web.config ra, sẽ thấy cặp thẻ thêm mới vào với thuộc tính mode là Forms

```
1 <?xml version="1.0"?>
2 <configuration>
3   <connectionStrings>
4     <add name="sproConn" connectionString="Data Source=.;Database=SMS;providerName=System.Data.SqlClient" />
5   </connectionStrings>
6   <system.web>
7     <authentication mode="Forms">
8     </authentication>
9     <sessionState mode="InProc" cookieless="false" timeout="20">
10    </sessionState>
11  </system.web>
12 </configuration>
```

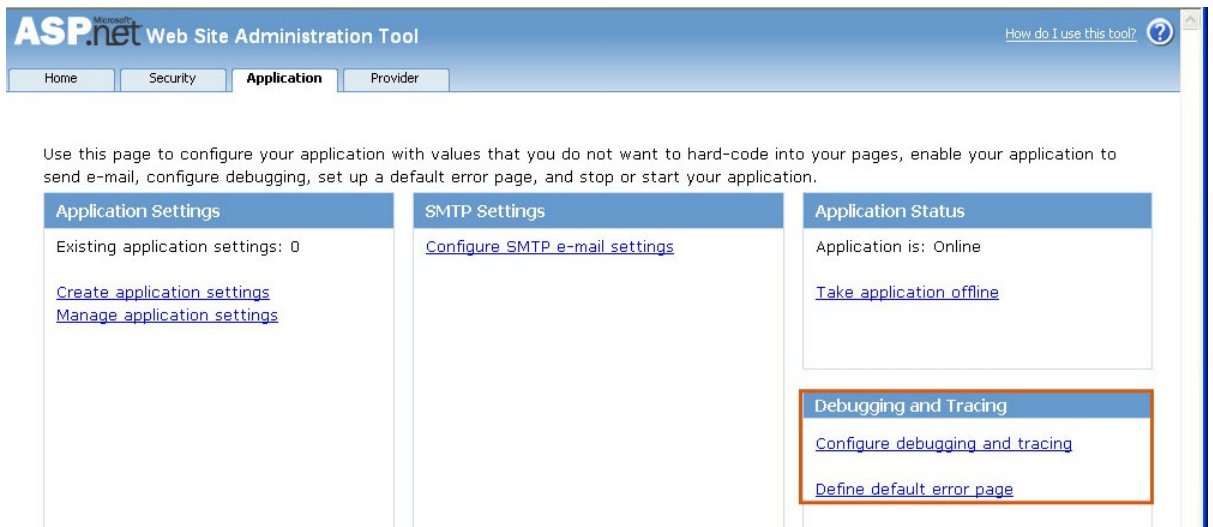
14. Trong cặp thẻ authentication, bạn hãy cấu hình thông tin form đăng nhập như sau:

```
1 <?xml version="1.0"?>
2 <configuration>
3   <connectionStrings>
4     <add name="sproConn" connectionString="Data Source=.;Database=SMS;providerName=System.Data.SqlClient"/>
5   </connectionStrings>
6   <system.web>
7     <authentication mode="Forms">
8       <forms name=".ASPXAUTH" loginUrl="Login.aspx" timeout="30" path="/" slidingExpiration="true">
9       </forms>
10    </authentication>
11    <sessionState mode="InProc" cookieless="false" timeout="20">
12    </sessionState>
13  </system.web>
14 </configuration>
```

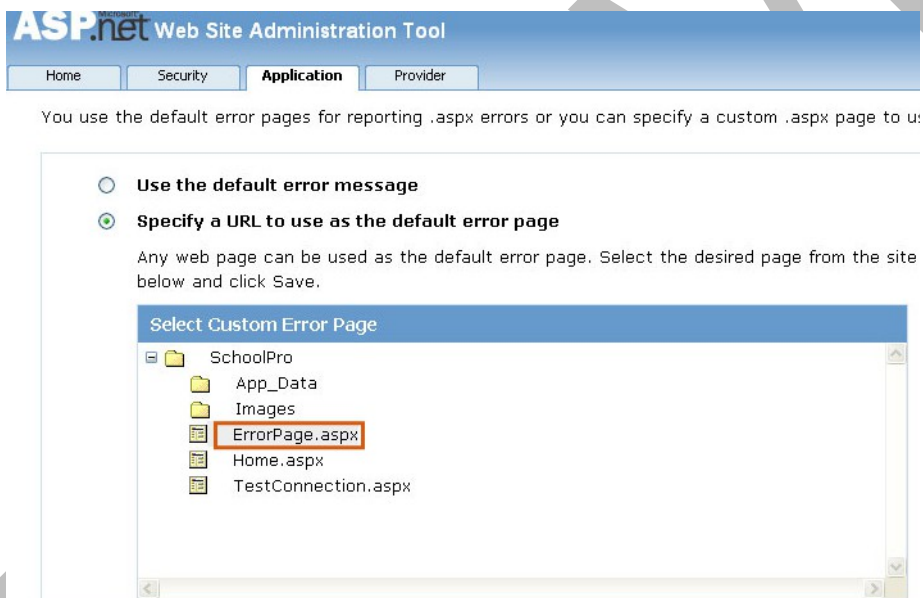
15. Thêm mới một Web Form vào dự án và đặt tên là **ErrorPage.aspx**

16. Chuyển về chế độ thiết kế và nhập một thông điệp “The server is busy!, Try again please”.

17. Trong cửa sổ của trang cấu hình (Website Administration tool) chọn Tab **Application**



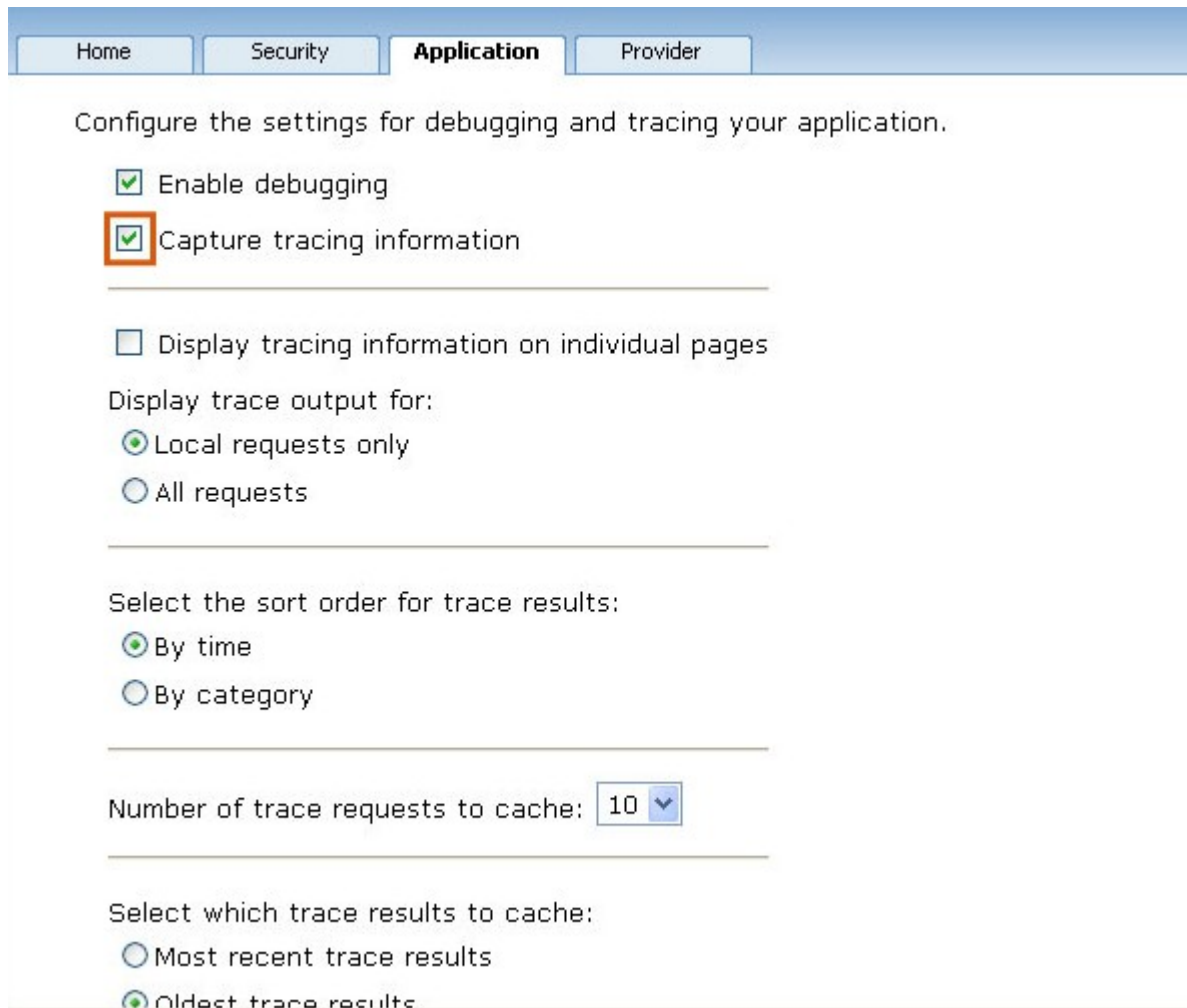
18. Bấm chọn **Define default error page**



19. Bấm nút **Save** và bấm nút **OK**

20. Mở file Web.config, bạn sẽ thấy thông tin cấu hình mới được thêm vào.

21. Tương tự từ bước 17, bạn bấm chọn **Configure debugging and tracing** và chọn cấu hình như hình dưới đây.



Home Security **Application** Provider

Configure the settings for debugging and tracing your application.

☒ Enable debugging

☒ Capture tracing information

☐ Display tracing information on individual pages

Display trace output for:

☒ Local requests only

☐ All requests

Select the sort order for trace results:

☒ By time

☐ By category

Number of trace requests to cache: 10

Select which trace results to cache:

☐ Most recent trace results

☒ Oldest trace results

Bài thực hành lập trình ASP.NET nâng cao – Bài 1-2

Trong bài thực hành này bạn sẽ thực hiện các thao tác cấu hình bảo mật Web theo các yêu cầu dưới đây

- Khi người dùng truy cập vào trang Default.aspx sẽ không cần đăng nhập
- Khi người dùng truy cập vào trang trong thư mục Admin yêu cầu phải đăng nhập

22. Mở ứng dụng Web từ thư mục **ASPNET_ADV_Lab12** chỉ định

23. Trong **Solution Explorer** , bấm chuột phải vào file **Default.aspx**, và chọn **View in Browser** bạn sẽ thấy thông tin của trang

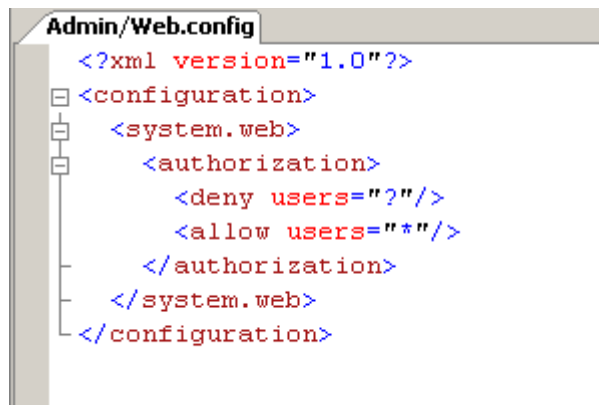
24. Trong **Solution Explorer** , bấm chuột phải vào file **AdminPage.aspx**, và chọn **View in Browser** bạn sẽ thấy thông tin của trang mà không bị yêu cầu đăng nhập.

25. Trong **Solution Explorer** thêm mới một file **Web.config** vào trong thư mục gốc của ứng dụng Web. Và nhập thông tin cấu hình như hình dưới đây:



```
<?xml version="1.0"?>
<configuration>
  <system.web>
    <authentication mode="Forms">
      <forms name="frmLogin" loginUrl="login.aspx" />
    </authentication>
  </system.web>
</configuration>
```

26. Trong **Solution Explorer** thêm mới một file **Web.config** vào trong thư mục **Admin** của ứng dụng Web. Và nhập thông tin cấu hình như hình dưới đây:



```
<?xml version="1.0"?>
<configuration>
  <system.web>
    <authorization>
      <deny users="?" />
      <allow users="*" />
    </authorization>
  </system.web>
</configuration>
```

27. Trong **Solution Explorer** , bấm chuột phải vào file **Default.aspx**, và chọn **View in Browser** bạn sẽ thấy thông tin của trang không phải đăng nhập
28. Trong **Solution Explorer** , bấm chuột phải vào file **AdminPage.aspx**, và chọn **View in Browser** bạn sẽ được yêu cầu đăng nhập cấu đăng nhập. Hãy nhập vào user name là **admin** và mật khẩu là **password** và bấm nút **Login** và kiểm tra kết quả
29. Trong **Solution Explorer** , bấm chuột phải vào file **Login.aspx**, và chọn **View Code** xem đoạn mã và nhận xét
30. Trong **Solution Explorer** , bấm chuột phải vào file **AdminPage.aspx**, và chọn **View Code** xem đoạn mã và nhận xét

ASPNET_ADV_Lab3

Trong bài thực hành này bạn sẽ thực hiện các thao tác cấu hình và sử dụng profile theo các yêu cầu dưới đây

1. Mở ứng dụng Web từ thư mục **ASPNET_ADV_Lab3** chỉ định
2. Copy 2 tệp cơ sở dữ liệu trong bài tập 2-1 vào thư mục App_Data
3. Trong **Solution Explorer** bấm đúp vào file Web.config để mở file
4. Nhập thông tin cấu hình cấu trúc profile như sau:

```
<profile>
  <properties>
    <add name="FirstName"/>
    <add name="LastName"/>
    <add name="Age"/>
    <add name="Time"/>
    <group name="FamilyDetails">
      <add name="MaritalStatus"/>
      <add name="NumberOfChildren"/>
    </group>
  </properties>
</profile>
```

5. Hãy đưa ra nhận xét thông tin cấu trúc bạn vừa cấu hình?
6. Trong **Solution Explorer** mở file **ProfilePage.aspx** ở chế độ thiết kế.
7. Xem và nhận xét về thông tin cũng như giao diện của trang
8. Bấm chuột phải lên **ProfilePage.aspx** và chọn **View Code**
9. Trong sự kiện Page_Load nhập đoạn mã sau:

```
lblTime.Text = DateTime.Now.ToString();
```

10. Nhận xét đoạn mã trên?
11. Chuyển **ProfilePage.aspx** về chế độ thiết kế.
12. Bấm đúp vào nút **Save Values to Profile**
13. Viết mã cho sự kiện như sau:

```
protected void btnSaveProfile_Click(object sender, EventArgs e)
{
    Profile.FirstName = txtFirstName.Text;
    Profile.LastName = txtLastName.Text;
    Profile.Age = txtAge.Text;
    Profile.FamilyDetails.MaritalStatus = txtMaritalStatus.Text;
    Profile.FamilyDetails.NumberOfChildren = txtNoOfChildren.Text;
    Profile.Time = lblTime.Text;
}
```

14. Chuyển **ProfilePage.aspx** về chế độ thiết kế.
15. Bấm đúp vào nút **Fetch Values from Profile**

16. Viết mã cho sự kiện như sau:

```
protected void btnFetchProfile_Click(object sender, EventArgs e)
{
    panProfile.Visible = true;
    lblShowFirstName.Text = Profile.FirstName;
    lblShowLastName.Text = Profile.LastName;
    lblShowAge.Text = Profile.Age;
    lblShowMaritalStatus.Text = Profile.FamilyDetails.MaritalStatus;
    lblShowNoOfChildren.Text = Profile.FamilyDetails.NumberOfChildren;
    lblShowTime.Text = Profile.Time;
}
```

17. Bạn hãy đưa ra nhận xét cho đoạn mã.

18. Trong **Solution Explorer** bấm chuột phải file **ProfilePage.aspx** và chọn **View in Browser**.

19. Nhập thông tin và bấm vào nút **Save Values to Profile**

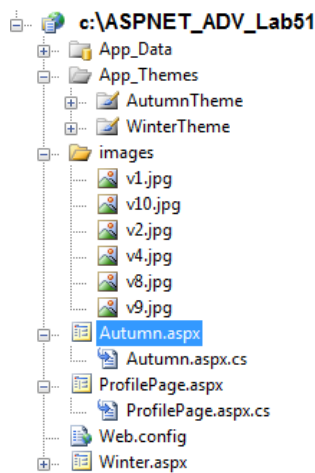
20. Bấm tiếp vào nút **Fetch Values from Profile**

21. Nhận xét kết quả

Bài thực hành lập trình thiết kế web bằng ASP.NET nâng cao – Bài 4

Trong bài thực hành này bạn sẽ thực hiện các thao tác cấu hình và sử dụng Themes theo các yêu cầu dưới đây

1. Mở ứng dụng Web từ thư mục **ASPNET_ADV_Lab4** trong bài thực hành trước. Nếu đang mở thì bước này bỏ qua.
2. Trong **Solution Explorer** bấm chuột phải vào dự án và chọn **Add ASP.NET Folder** và chọn **App_Themes**
3. Mở thư mục **ASPNET_ADV_Lab4** và copy 2 thư mục là **AutumnTheme** và **WinterTheme** dán vào trong thư mục **App_Themes**
4. Mở thư mục **ASPNET_ADV_Lab4** và copy thư mục **images** và các file còn lại và dán vào thư mục gốc của ứng dụng Web
5. Cấu trúc dự án của bạn như sau:



6. Mở rộng thư mục **AutumnTheme** và bấm đúp vào file **Autumn.skin** xem và nhận xét thông tin trong file.
7. Mở rộng thư mục **WinterTheme** và bấm đúp vào file **winter.skin** xem và nhận xét thông tin trong file.
8. Bấm đúp vào file **Autumn.aspx** và chuyển về chế độ thiết kế
9. Xem kỹ giao diện và đưa ra nhận xét?
10. Chuyển file **Autumn.aspx** về chế độ **Source** (html)
11. Trên dòng đầu tiên bạn hãy cấu hình kích hoạt Themes như sau;

```
<%@ Page Language="C#" AutoEventWireup="true" Theme="AutumnTheme" CodeFile="Autumn.aspx.cs" Inherits="Autumn" %>
```

12. Bấm đúp vào file **Winter.aspx** và chuyển về chế độ thiết kế
13. Xem kỹ giao diện và đưa ra nhận xét?
14. Chuyển file **Winter.aspx** về chế độ **Source** (html)
15. Trên dòng đầu tiên bạn hãy cấu hình kích hoạt Themes như sau;

```
<%@ Page Language="C#" AutoEventWireup="true" Theme="AutumnTheme" CodeFile="Autumn.aspx.cs" Inherits="Autumn" %>
```

Bấm đúp vào file **Autumn.aspx** và chuyển về chế độ thiết kế

16. Xem kỹ giao diện và đưa ra nhận xét?
17. Chuyển file **Autumn.aspx** về chế độ **Source** (html)
18. Trên dòng đầu tiên bạn hãy cấu hình kích hoạt Themes như sau;

```
<%@ Page Language="C#" AutoEventWireup="true" Theme="WinterTheme" CodeFile="Winter.aspx.cs" Inherits="Winter" %>
```

19. Chuyển file **Autumn.aspx** về chế độ thiết kế
20. Bấm chuột phải lên file **Autumn.aspx** và chọn **View Code**

21. Trong sự kiện Page_Load nhập đoạn mã sau

```
// Hiển thị theme hiện hành của trang trên label.
```

```
lblThemeName.Text = Page.Theme;
```

22. Bấm đúp vào nút **Button** và viết mã lệnh sau:

```
Response.Redirect("Winter.aspx");
```

23. Chuyển file **Winter.aspx** về chế độ thiết kế

24. Bấm chuột phải lên file **Winter.aspx** và chọn **View Code**

25. Trong sự kiện Page_Load nhập đoạn mã sau

```
// Hiển thị theme hiện hành của trang trên label.
```

```
lblThemeName.Text = Page.Theme;
```

26. Bấm đúp vào nút **Button** và viết mã lệnh sau:

```
Response.Redirect("Autumn.aspx");
```

27. Bấm chuột phải lên file **Autumn.aspx** chọn **View in Browser**

28. Bấm vào nút Button và kiểm tra kết quả.

29. Bấm đúp vào file **Autumn.skin**

30. Thay đổi thuộc tính **ForeColor** theo màu bạn thích

31. Bấm đúp vào file **winter.skin**

32. Thay đổi thuộc tính **ForeColor** theo màu bạn thích

33. Bấm chuột phải lên file **Autumn.aspx** chọn **View in Browser**

34. Nhận xét kết quả đạt được.

Bài thực hành lập trình thiết kế web bằng ASP.NET nâng cao – Bài 5

Trong bài thực hành này bạn sẽ thực hiện các yêu cầu dưới đây

- Tạo một ứng dụng Web cho việc toàn cầu hóa.
- Build và chạy ứng dụng Web.

1. Tạo một ứng dụng Web đặt tên là Lab5.

2. Xóa file **default.aspx** mặc định.

3. Thêm mới một Web Form vào dự án và đặt tên là **Exchange.aspx**

4. Chuyển Web Form về chế độ thiết kế.

5. Thực hiện thiết kế Web Form giao diện tương tự như hình dưới đây với các điều kiện được mô tả trong bảng.

The screenshot shows a web browser window with the address bar displaying 'Exchange.aspx' and 'Start Page'. The page title is 'Welcome to Ultra Trade Center'. Below the title, there is a 'Select Language:' label followed by a dropdown menu with the text 'Select a language'. Below this, there is an 'Exchange Rate:' label followed by a text box containing '[lblMoneyConvert]'. At the bottom, there is a calendar for 'Tháng Hai 2007' (February 2007) with days of the week (H, B, T, N, S, B, C) and dates (29, 30, 31, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31).

Bảng mô tả các điều khiển sử dụng cho giao diện

| STT | Controls | Thuộc tính và giá trị của thuộc tính |
|-----|--------------|--|
| 1. | Lable | <ul style="list-style-type: none"> - ID: lblHeader - Text: Welcome to Ultra Trade Center - Font: Verdana, Large - ForeColor: Blue - Width: 591px - Height: 23px |
| 2. | Lable | <ul style="list-style-type: none"> - ID: lblLanguage - Width: 136px - Text: Select Language: |
| 3. | DropDownList | <ul style="list-style-type: none"> - ID: ddlLanguage - AutoPostBack: True - Items: <ul style="list-style-type: none"> + Text: Select a language ; Value: -1 + Text: US English; Value: 1 + Text: French; Value: 2 |
| 4. | Lable | <ul style="list-style-type: none"> - ID : lblMoneyConvert - Text: ĐỂ trống - Width: 152px - Height: 40px |
| 5. | Calendar | <ul style="list-style-type: none"> - ID : calCalendar - Width: 305px - Height: 228px - ShowGridLines: True - BackColor: #FFFFCC |

| | | |
|--|--|---|
| | | <ul style="list-style-type: none"> - BorderColor: #FFCC66 - DayNameFormat: Shortest - Font: Verdana, 8pt - ForeColor: #663399 |
|--|--|---|

- Trong cửa sổ Solution Explore bấm chuột phải lên file **Exchange.aspx** và chọn View Code
- Bên dưới sự kiện **Page_Load**, xây dựng một phương thức xác định Culture người dùng chọn như sau:

```
public void DoAction(string text)
{
    string strCur1 = "1 Dollar = ";
    string strCur2 = " Euro";
    double dMoney = 1;
    switch (text)
    {
        case "1": SetCulture("en-US", "en-US");
                dMoney = (dMoney * 0.62775);
                break;
        case "2": SetCulture("en-GB", "en-GB");
                dMoney = (dMoney * 0.62775);
                break;
        case "3": SetCulture("fr-FR", "fr-FR");
                dMoney = (dMoney / 0.62775);
                strCur1 = "1 Euro = ";
                strCur2 = " Dollars";
                break;
        default: break;
    }
    if (Session["CurUICulture"] != null && Session["CurCulture"] != null)
    {
        Thread.CurrentThread.CurrentUICulture = (CultureInfo)Session["CurUICulture"];
        Thread.CurrentThread.CurrentCulture = (CultureInfo)Session["CurCulture"];
    }
    lblMoneyConvert.Text = strCur1 + dMoney + strCur2;
}
```

- Bên dưới sự kiện phương thức DoAction xây dựng một phương thức thực hiện gán Culture như sau.

```
protected void SetCulture(string name, string locale)
{
    Thread.CurrentThread.CurrentUICulture = new CultureInfo(name);
    Thread.CurrentThread.CurrentCulture = new CultureInfo(locale);
    Session["CurUICulture"] = Thread.CurrentThread.CurrentUICulture;
    Session["CurCulture"] = Thread.CurrentThread.CurrentCulture;
}
```

- Chuyển Web Form về chế độ thiết kế
- Bấm đúp vào DropDownList control và viết mã lệnh cho sự kiện **ddlLanguage_SelectedIndexChanged** như sau:

```
protected void ddlLanguage_SelectedIndexChanged(object sender, EventArgs e)
{
    DoAction(ddlLanguage.SelectedValue);
}
```

- Ghi lại toàn bộ dự án

12. Bấm chuột phải lên **Exchange.aspx** và chọn **View in Browser**
13. Trong DropDownList control thay đổi ngôn ngữ
14. Nhận xét kết quả nhận được
15. Bạn có thể thêm một vài culture khác mà bạn muốn sau đó thử lại chương trình.

Bài thực hành lập trình thiết kế web bằng ASP.NET nâng cao – Bài 6

Trong bài thực hành này bạn sẽ thực hiện các yêu cầu dưới đây

- Mở ứng dụng trong bài Lab5.
 - Tạo Resource File cho ứng dụng Web
 - Tạo Localization cho ứng dụng Web sử dụng Resource file
 - Build và chạy ứng dụng
16. Mở ứng dụng Web Lab5 đã tạo trong bài trước. Nếu đang mở bỏ qua bước này.
 17. Chuyển file **Exchange.aspx** về chế độ thiết kế.
 18. Bấm chọn vào **lblHeader** control, trong cửa sổ thuộc tính xóa trắng thuộc tính Text
 19. Thêm hai **Label** control vào bên dưới **lblMoneyConvert** control với thông tin như sau:
 - ID: **lblSelectDate**
 - Text: **Select a Date:**
 - Width: **187px**
 - ID: **lblSelectedDate**
 - Visible :**False**
 - Text: **Để trống**
 20. Kết quả giao diện lúc này như sau:

[lblHeader]

Select Language:

Exchange Rate:

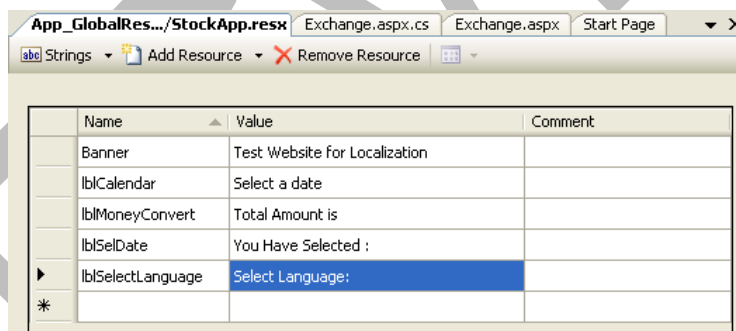
[lblMoneyConvert]

Select a Date:

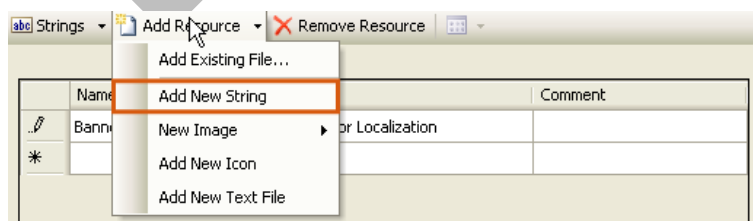
[lblSelectedDate]

| < Tháng Hai 2007 > | | | | | | |
|--------------------|----|----|----|----|----|----|
| H | B | T | N | S | B | C |
| 29 | 30 | 31 | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| 26 | 27 | 28 | 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 | 9 | 10 | 11 |

21. Bấm chuột phải lên dự án và chọn **Add ASP.NET Folder** và chọn **App_GlobalResources**
22. Bấm chuột phải lên thư mục **App_GlobalResources** và chọn **Add New Item**
23. Trong cửa sổ **Add New Item**, chọn **Resource file** và nhập tên là **StockApp.resx**
24. Bấm đúp vào file **StockApp.resx** để mở
25. Hãy nhập đầy đủ thông tin như hình dưới đây:



Lưu ý: Để thêm một Item bạn chọn **Add Resource** menu phía trên



26. Sau khi hoàn tất bước trên. Bấm chuột phải lên file **StockApp.resx** và chọn lệnh **Copy**.
27. Bấm chuột phải lên thư mục **App_GlobalResources** và chọn **Paste**

28. Bạn hãy đổi tên file vừa copy thành **StockApp.fr-FR.resx** (fr-FR là culture Pháp)

29. Bấm đúp vào file vừa được copy để sửa ngôn ngữ. Bạn hãy sửa ngôn ngữ như sau:

| | Name | Value | Comment |
|---|-------------------|--|---------|
| | Banner | Bienvenue pour commercer ultra le centre | |
| | lblCalendar | Choisissez une date | |
| | lblMoneyConvert | Vous Avez choisi : | |
| | lblSelDate | Choisissez une date: | |
| ▶ | lblSelectLanguage | Choisissez La Langue: | |
| * | | | |

30. Tương tự từ bước 11 đến bước 14 hãy thực hiện với các ngôn ngữ khác nếu bạn muốn.

31. Bấm chuột phải lên file **Exchange.aspx** và chọn View Code.

32. Trong sự kiện Page_Load bạn hãy viết lệnh sau:

```
protected void Page_Load(object sender, EventArgs e)
{
    if (!Page.IsPostBack)
    {
        CultureInfo objCultureInfo = Thread.CurrentThread.CurrentCulture;
        LoadResources(objCultureInfo);
    }
}
```

33. Bên dưới sự kiện Page_Load xây dựng một phương thức **LoadResources** để truy cập tới Resource file. Mã như sau:

```
public void LoadResources(CultureInfo objCultureInfo)
{
    String strResourcesPath = Server.MapPath("App_GlobalResources");
    ResourceManager objResourceMgr = new ResourceManager("Resources.StockApp", Assembly.Load("App_GlobalResources"));
    lblHeader.Text = objResourceMgr.GetString("Banner", objCultureInfo);
    lblLanguage.Text = objResourceMgr.GetString("lblSelectLanguage", objCultureInfo);
    lblSelectedDate.Text = objResourceMgr.GetString("lblSelDate", objCultureInfo);
    lblSelectDate.Text = objResourceMgr.GetString("lblSelectDate", objCultureInfo);
}
```

34. Bên trong phương thức **DoAction** thêm đoạn mã được đánh dấu dưới đây

```
CultureInfo objCultureInfo = Thread.CurrentThread.CurrentCulture;
LoadResources(objCultureInfo);
lblSelectedDate.Text += " : " + calCalendar.SelectedDate.Date.ToShortDateString();
iDay = calCalendar.SelectedDate.Day;
if (Session["CurUICulture"] != null && Session["CurCulture"] != null)
{
    Thread.CurrentThread.CurrentUICulture = (CultureInfo)Session["CurUICulture"];
    Thread.CurrentThread.CurrentCulture = (CultureInfo)Session["CurCulture"];
}
```

```
if (calCalendar.SelectedDate <= DateTime.Today)
{
    lblMoneyConvert.Text = strCur1 + dMoney + strCur2;
}
else
{
    lblMoneyConvert.Text = "Money value cannot be shown for future dates";
}
```

35. Trong chế độ thiết kế, bấm đúp vào Calendar control và viết sự kiện sau:

```
protected void calCalendar_SelectionChanged(object sender, EventArgs e)
{
    lblSelectedDate.Visible = true;
    DoAction(ddlLanguage.SelectedValue);
}
```

36. Trong Solution Explore bấm chuột phải lên **Exchange.aspx** và chọn **View in Browser**.

37. Thay đổi ngôn ngữ trong **DropDownList** control để kiểm tra kết quả.

38. Thay đổi ngày tháng trong Calendar control.

39. Hãy nhận xét các bước vừa thực hiện

Bài thực hành lập trình thiết kế web bằng ASP.NET nâng cao – Bài 7-1

Trong bài thực hành này bạn sẽ thực hiện các yêu cầu dưới đây

- Tạo một Web User control.
- Thêm Web User control vào trong Web Form
- Build và chạy chương trình

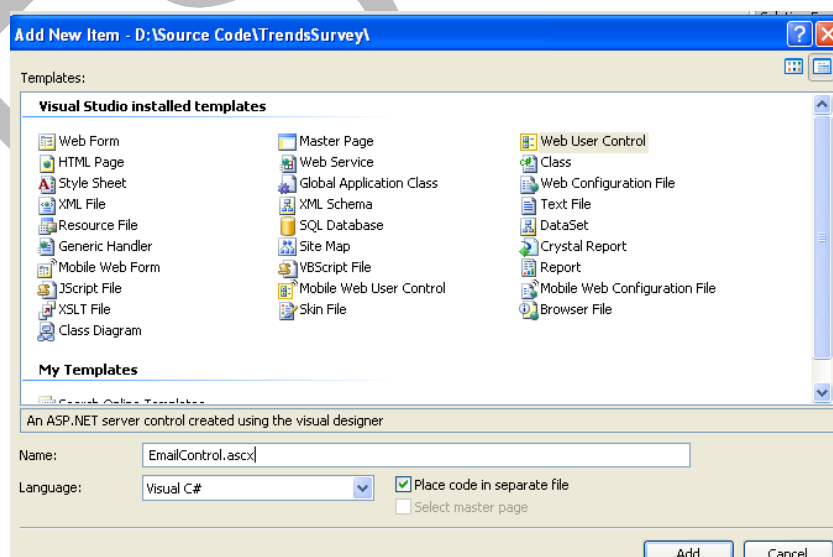
40. Tạo một ứng dụng Web và đặt tên là Lab7_1

41. Bấm chuột phải lên dự án và chọn Add New Item

42. Trong cửa sổ Add New Item chọn Web User Control

43. Trong hộp Name nhập tên control là **EmailControl.ascx**

44. Chọn ngôn ngữ C#. Kết quả như hình dưới đây:



45. Bấm nút Add để thêm vào dự án.

46. Mở file **EmailControl.ascx** ở chế độ HTML code. Hãy nhận xét dòng đầu tiên của trang.

47. Chuyển trang về chế độ thiết kế

48. Xây dựng giao diện cho control như hình dưới đây:

49. Hãy đặt tên control và các thuộc tính cho control như bảng dưới đây:

| STT | Controls | Thuộc tính và giá trị của thuộc tính |
|-----|----------|---|
| 6. | Lable | - ID: lblTitle - Text: ĐỂ TRẮNG - Width: 353px |
| 7. | Lable | - ID: lblEmailID - Width: 95px - Text: Your Email ID: |
| 8. | TextBox | - ID: txtEmailID - Width: 166px |
| 9. | Lable | - ID : lblComments - Text: Comments: - Width: 96px |
| 10. | TextBox | - ID : txtComments - TextMode: MultiLine - Height: 75px - Width: 357px |
| 11. | Button | - ID: btnSubmit - Text: Submit - Width: 78px |

50. Bấm đúp vào nút **Submit** và viết mã lệnh sau:

```
protected void btnSubmit_Click(object sender, EventArgs e)
{
    MailMessage objMailMessage = new MailMessage();
    objMailMessage.To.Add(new MailAddress("trends@hotmail.com"));
    objMailMessage.From = new MailAddress(txtEmailID.Text);
    objMailMessage.Body = txtComments.Text;
    SmtpClient objSmtp = new SmtpClient();
    objSmtp.Send(objMailMessage);
    Response.Write("Thank you. Your comments are sent.");
}
```

Lưu ý: thêm Namespace sau vào đầu trang mã để thực hiện gửi mail.

using System.Net.Mail;

51. Bên dưới sự kiện **btnSubmit_Click** viết thêm một thuộc tính như sau:

```
public string Text
{
    get
    {
        return this.lblTitle.Text;
    }
    set
    {
        lblTitle.Text = value;
    }
}
```

52. Thêm mới một Web Form vào dự án và đặt tên là **Feedback.aspx**.

53. Chuyển Web Form về chế độ thiết kế.

54. Trong cửa sổ Solution Explorer, bấm chọn file **EmailControl.ascx** và kéo thả vào Web Form **Feedback.aspx**

55. Chuyển Web Form **Feedback.aspx** về chế độ HTML Source và để ý dòng thứ hai của trang mã HTML. Đưa ra nhận xét

56. Để ý Control được thêm vào ở dạng HTML code.

57. Ghi toàn bộ dự án lại.

58. Bấm chuột phải lên trang **Feedback.aspx** chọn View in Browser bạn sẽ thấy control do bạn tạo được hiển thị trong trang Web.

Bài thực hành lập trình thiết kế web bằng ASP.NET nâng cao – Bài 7-2

Trong bài thực hành này bạn sẽ thực hiện các yêu cầu dưới đây

- Tạo và sử dụng một Custom Web Server Control.

59. Tạo một ứng dụng Web và đặt tên là Lab7_2
60. Thêm thư mục App_Code vào dự án
61. Bấm chuột phải vào thư mục App_Code và thêm mới một Class.
62. Đặt tên file là MyCustomControl.cs
63. Nhập đoạn mã để tạo control như sau

```
using System;
using System.ComponentModel;
using System.Drawing;
using System.Security.Permissions;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Samples.AspNet.CS.Controls
{
    [
        AspNetHostingPermission(SecurityAction.Demand,
            Level = AspNetHostingPermissionLevel.Minimal),
        AspNetHostingPermission(SecurityAction.InheritanceDemand,
            Level = AspNetHostingPermissionLevel.Minimal),
        DefaultEvent("Submit"),
        DefaultProperty("ButtonText"),
        ToolboxData("<{0}:Register runat='server'> </{0}:Register>"),
    ]
    public class Register : CompositeControl
    {
        private Button submitButton;
        private TextBox nameTextBox;
        private Label nameLabel;
        private TextBox emailTextBox;
        private Label emailLabel;
        private RequiredFieldValidator emailValidator;
        private RequiredFieldValidator nameValidator;

        private static readonly object EventSubmitKey =
            new object();

        // The following properties are delegated to
        // child controls.
        [
            Bindable(true),
            Category("Appearance"),
            DefaultValue(""),
            Description("The text to display on the button.")
        ]
        public string ButtonText
        {
            get
            {
                EnsureChildControls();
                return submitButton.Text;
            }
        }
    }
}
```



```

        set
        {
            EnsureChildControls();
            submitButton.Text = value;
        }
    }

    [
        Bindable(true),
        Category("Default"),
        DefaultValue(""),
        Description("The user name.")
    ]
    public string Name
    {
        get
        {
            EnsureChildControls();
            return nameTextBox.Text;
        }
        set
        {
            EnsureChildControls();
            nameTextBox.Text = value;
        }
    }

    [
        Bindable(true),
        Category("Appearance"),
        DefaultValue(""),
        Description(
            "Error message for the name validator.")
    ]
    public string NameErrorMessage
    {
        get
        {
            EnsureChildControls();
            return nameValidator.ErrorMessage;
        }
        set
        {
            EnsureChildControls();
            nameValidator.ErrorMessage = value;
            nameValidator.ToolTip = value;
        }
    }

    [
        Bindable(true),
        Category("Appearance"),
        DefaultValue(""),
        Description("The text for the name label.")
    ]
    public string NameLabelText
    {

```

```

    get
    {
        EnsureChildControls();
        return nameLabel.Text;
    }
    set
    {
        EnsureChildControls();
        nameLabel.Text = value;
    }
}

[
    Bindable(true),
    Category("Default"),
    DefaultValue(""),
    Description("The e-mail address.")
]
public string Email
{
    get
    {
        EnsureChildControls();
        return emailTextBox.Text;
    }
    set
    {
        EnsureChildControls();
        emailTextBox.Text = value;
    }
}

[
    Bindable(true),
    Category("Appearance"),
    DefaultValue(""),
    Description(
        "Error message for the e-mail validator.")
]
public string EmailErrorMessage
{
    get
    {
        EnsureChildControls();
        return emailValidator.ErrorMessage;
    }
    set
    {
        EnsureChildControls();
        emailValidator.ErrorMessage = value;
        emailValidator.ToolTip = value;
    }
}

[
    Bindable(true),
    Category("Appearance"),

```

```

DefaultValue(""),
Description("The text for the e-mail label.")
]
public string EmailLabelText
{
    get
    {
        EnsureChildControls();
        return emailLabel.Text;
    }
    set
    {
        EnsureChildControls();
        emailLabel.Text = value;
    }
}

// The Submit event.
[
    Category("Action"),
    Description("Raised when the user clicks the button.")
]
public event EventHandler Submit
{
    add
    {
        Events.AddHandler(EventSubmitKey, value);
    }
    remove
    {
        Events.RemoveHandler(EventSubmitKey, value);
    }
}

// The method that raises the Submit event.
protected virtual void OnSubmit(EventArgs e)
{
    EventHandler SubmitHandler =
        (EventHandler)Events[EventSubmitKey];
    if (SubmitHandler != null)
    {
        SubmitHandler(this, e);
    }
}

// Handles the Click event of the Button and raises
// the Submit event.
private void _button_Click(object source, EventArgs e)
{
    OnSubmit(EventArgs.Empty);
}

protected override void RecreateChildControls()
{
    EnsureChildControls();
}

```

```

protected override void CreateChildControls()
{
    Controls.Clear();

    nameLabel = new Label();

    nameTextBox = new TextBox();
    nameTextBox.ID = "nameTextBox";

    nameValidator = new RequiredFieldValidator();
    nameValidator.ID = "validator1";
    nameValidator.ControlToValidate = nameTextBox.ID;
    nameValidator.Text = "Failed validation.";
    nameValidator.Display = ValidatorDisplay.Static;

    emailLabel = new Label();

    emailTextBox = new TextBox();
    emailTextBox.ID = "emailTextBox";

    emailValidator = new RequiredFieldValidator();
    emailValidator.ID = "validator2";
    emailValidator.ControlToValidate =
        emailTextBox.ID;
    emailValidator.Text = "Failed validation.";
    emailValidator.Display = ValidatorDisplay.Static;

    submitButton = new Button();
    submitButton.Text = "Submit";
    submitButton.ID = "button1";
    submitButton.Click
        += new EventHandler(_button_Click);

    this.Controls.Add(nameLabel);
    this.Controls.Add(nameTextBox);
    this.Controls.Add(nameValidator);
    this.Controls.Add(emailLabel);
    this.Controls.Add(emailTextBox);
    this.Controls.Add(emailValidator);
    this.Controls.Add(submitButton);
}

protected override void Render(HtmlTextWriter writer)
{
    AddAttributesToRender(writer);

    writer.AddAttribute(
        HtmlTextWriterAttribute.Cellpadding,
        "1", false);
    writer.RenderBeginTag(HtmlTextWriterTag.Table);

    writer.RenderBeginTag(HtmlTextWriterTag.Tr);
    writer.RenderBeginTag(HtmlTextWriterTag.Td);
    nameLabel.RenderControl(writer);

```

```

writer.RenderEndTag();
writer.RenderBeginTag(HtmlTextWriterTag.Td);
nameTextBox.RenderControl(writer);
writer.RenderEndTag();
writer.RenderBeginTag(HtmlTextWriterTag.Td);
nameValidator.RenderControl(writer);
writer.RenderEndTag();
writer.RenderEndTag();

writer.RenderBeginTag(HtmlTextWriterTag.Tr);
writer.RenderBeginTag(HtmlTextWriterTag.Td);
emailLabel.RenderControl(writer);
writer.RenderEndTag();
writer.RenderBeginTag(HtmlTextWriterTag.Td);
emailTextBox.RenderControl(writer);
writer.RenderEndTag();
writer.RenderBeginTag(HtmlTextWriterTag.Td);
emailValidator.RenderControl(writer);
writer.RenderEndTag();
writer.RenderEndTag();

writer.RenderBeginTag(HtmlTextWriterTag.Tr);
writer.AddAttribute(
    HtmlTextWriterAttribute.Colspan,
    "2", false);
writer.AddAttribute(
    HtmlTextWriterAttribute.Align,
    "right", false);
writer.RenderBeginTag(HtmlTextWriterTag.Td);
submitButton.RenderControl(writer);
writer.RenderEndTag();
writer.RenderBeginTag(HtmlTextWriterTag.Td);
writer.Write("&nbsp;");
writer.RenderEndTag();
writer.RenderEndTag();

writer.RenderEndTag();
}
}
}

```

64. Xem kỹ nội dung mã lệnh và đưa ra nhận xét hoặc bổ sung thêm mã nếu muốn
65. Bấm chuột phải lên dự án và chọn Build Web Site.
66. Mở file **Default.aspx** ở chế độ HTML Code
67. Bên dưới dòng chỉ thị, đăng ký sử dụng control như sau :

Client Objects & Events (No Events)

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs" Inherits="_Default" %>
<% Register TagPrefix="MyControl" Namespace="Samples.AspNet.CS.Controls" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Untitled Page</title>
</head>
<body>
<form id="form1" runat="server">
<div>
</div>
</form>
</body>
</html>
```

68. Bên trong thẻ div hãy khai báo sử dụng control như sau:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
<title>Untitled Page</title>
</head>
<body>
<form id="form1" runat="server">
<div>
<MyControl:Register ID="ctrlLogin" runat="server" />
</div>
</form>
</body>
</html>
```

69. Chuyển trang về chế độ thiết kế.

70. Nhận xét giao diện.

71. Chạy và thử control.

72. Đưa ra nhận xét.

73. Tương tự bạn có thể tạo các control khác bạn muốn.