Mục lục

[II) Creating Named Skins. 3](#_Toc499251743)

[III) Themes Versus StyleSheetThemes. 4](#_Toc499251744)

[IV) Disabling Themes: 6](#_Toc499251745)

[V)Registering Themes in the Web Configuration File: 9](#_Toc499251746)

[VI)Adding Cascading Style Sheets to Themes: 11](#_Toc499251747)

[VII) Adding Multiple Cascading Style Sheets to a Theme: 13](#_Toc499251748)

[VIII)Changing Page Layouts with Cascading Style Sheets: 14](#_Toc499251749)

**I)Adding Skins to Themes**

Một Themes có thể chứa một hoặc nhiều tập tin Skin. Skin cho phép bạn sửa đổi bất kỳ các thuộc tính của một điều khiển ASP.NET có ảnh hưởng đến sự xuất hiện của nó. Một thư mục Theme có thể chứa một tệp Skin cho hàng trăm control. Ngoài ra, một Themes có thể chứa hàng trăm tệp Skin, mỗi tệp chứa một tệp đơn Skin. Nó không quan trọng bạn tổ chức Skins của bạn như thế nào vào các tập tin bởi vì tất cả mọi thứ trong một Thư mục Themes cuối cùng được biên dịch thành một lớp Themes.

Ví dụ, tưởng tượng rằng bạn quyết định rằng bạn muốn mọi điều khiển TextBox trong web của bạn ứng dụng xuất hiện với màu nền màu vàng và đường viền chấm. Nếu bạn thêm trong Ex1.1 cho Simple Theme (thư mục App\_Themes \ Simple), bạn có thể sửa đổi sự xuất hiện của tất cả các điều khiển TextBox trong tất cả các trang sử dụng Simple Theme.

<asp:TextBox

BackColor=”Yellow”

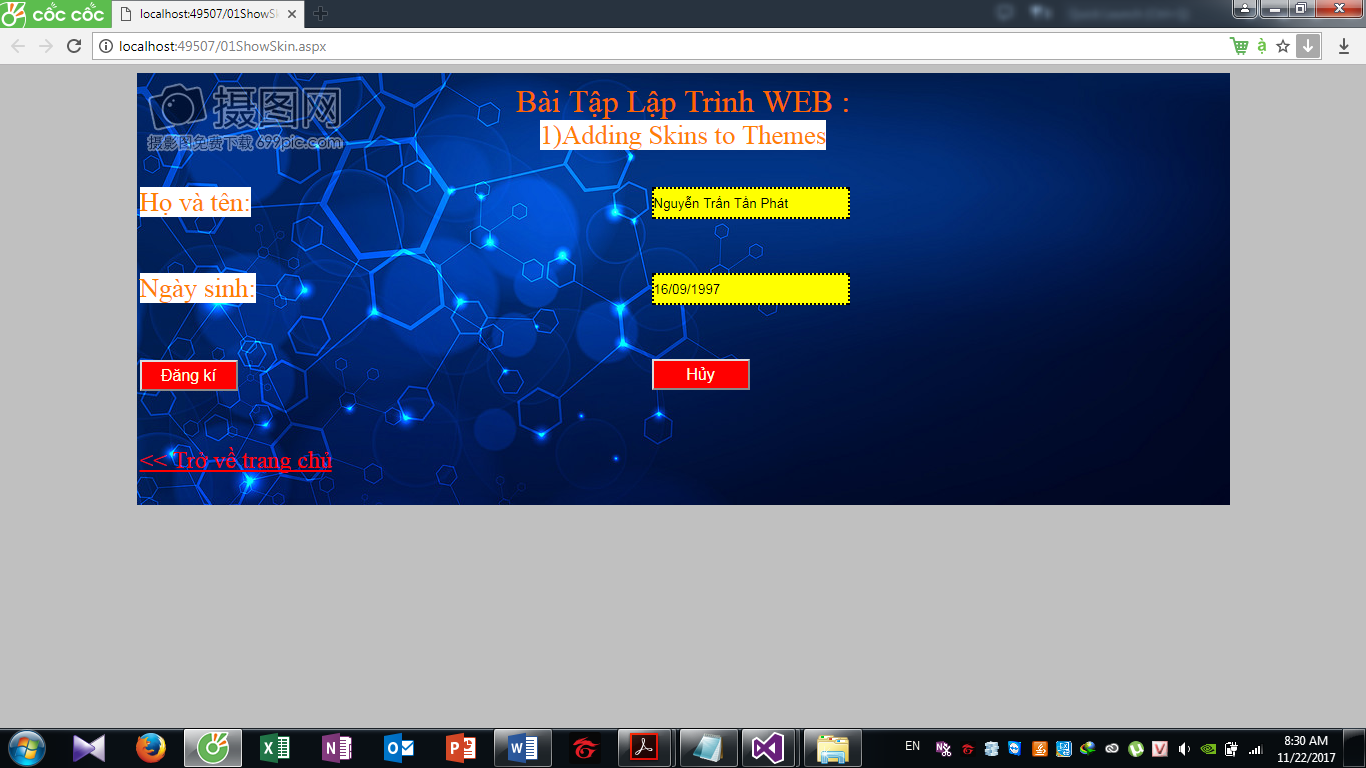
BorderStyle=”Dotted”

Runat=”Server” />

Skin được áp dụng cho mọi trang mà Simple Theme được áp dụng. Ví dụ, trong Ex1.2 sử dụng Simple Theme.

Ex1.2 ShowSkin.aspx

<%@ Page Language=”C#” Theme=”Simple” %>



# **II) Creating Named Skins.**

Trong phần trước, chúng tôi tạo ra một cái gì đó được gọi là Default Skin. Default Skin là áp dụng cho mọi trường hợp kiểm soát một loại nhất định. Ví dụ: Default Skin mặc định là áp dụng cho mọi trường hợp của một điều khiển TextBox.

Bạn cũng có thể chọn tạo Named Skin. Khi bạn tạo Named Skin, bạn có thể quyết định khi nào bạn muốn áp dụng Skin. Ví dụ: bạn có thể muốn các trường bắt buộc trong một hình thức để xuất hiện với một đường viền màu đỏ. Trong trường hợp đó, bạn có thể tạo một Named Skin và áp dụng Skin cho các điều khiển TextBox cụ thể. Skin trong Ex2 chứa cả Default Skin và Named Skin cho TextBox, Lable, Button.

Ex2:AppTheme\Simple2\simple2.skin

<asp:TextBox SkinID="DashedTextBox" BorderStyle="Dashed" BorderWidth="5px" Runat="Server" />

<asp:TextBox BorderStyle="Double" BorderWidth="5px" Runat="Server" />

<asp:Label runat="server" SkinID="Redlb" ForeColor="Red" Font-Size="20pt" BackColor="White"/>

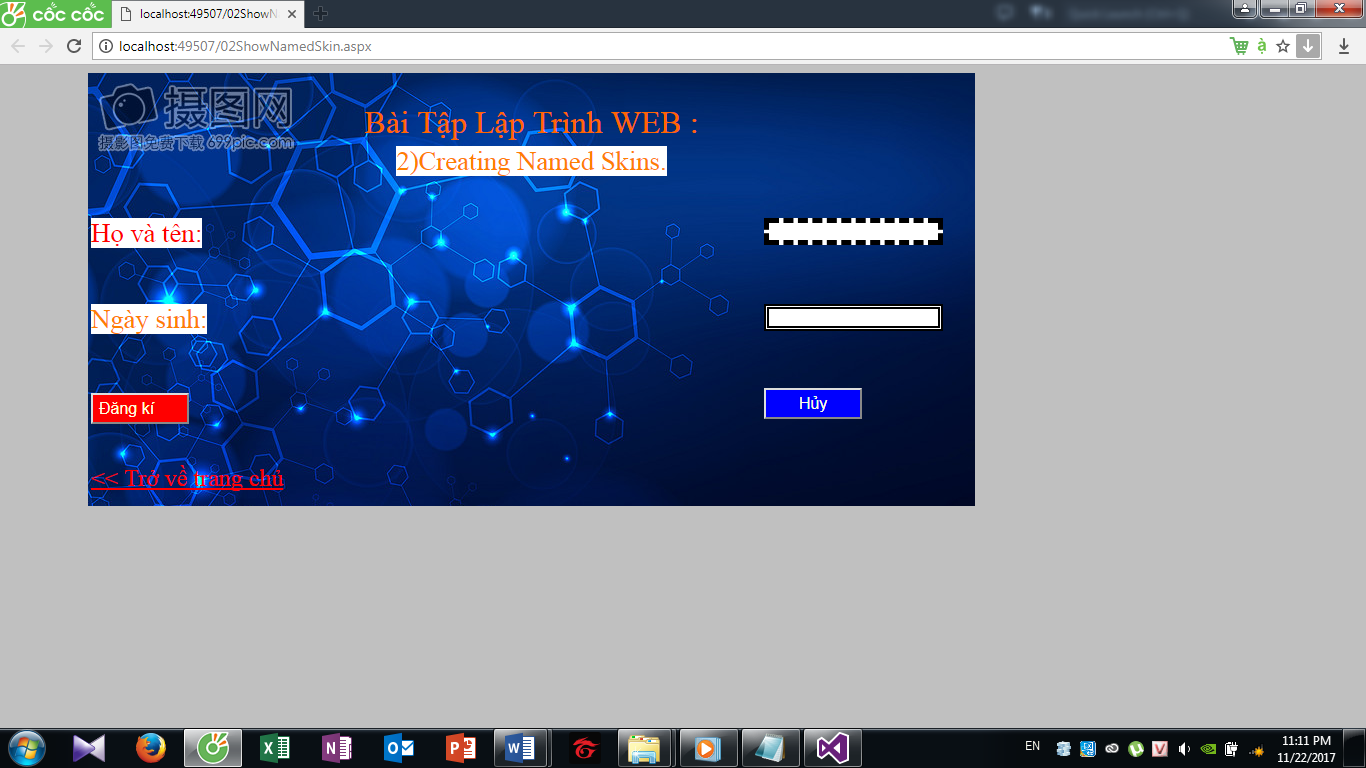
<asp:Label runat="server" ForeColor="#FF8000" Font-Size="20pt" BackColor="White"/>

<asp:Button runat="server" SkinID="Redbt" ForeColor="LightYellow" BackColor="Red" Font Size="12pt"/>

<asp:Button runat="server" ForeColor="LightYellow" BackColor="Blue" Font-Size="12pt"/>

Một Theme chỉ có thể chứa một Default Skin cho mỗi kiểu điều khiển. Tuy nhiên, Theme có thể chứa nhiều Named Skin như bạn muốn. Mỗi Named Skin đặt tên phải có một tên duy nhất.

Trang trong Ex2 chứa hai điều khiển TextBox. Điều khiển TextBox đầu tiên bao gồm một thuộc tính SkinID. Thuộc tính này làm cho Named Skin được áp dụng cho điều khiển. TextBox thứ hai, mặt khác, không bao gồm một thuộc tính SkinID. Default Skin được áp dụng cho điều khiển TextBox thứ hai tương tự với Button vá Lable.



# **III) Themes Versus StyleSheetThemes.**

Khi bạn áp dụng một chủ đề cho một trang, Skins trong Theme sẽ ghi đè lên bất kỳ tập tin nào hiện có, quan hệ của các điều khiển trong trang. Nói cách khác, thuộc tính trong Skin ghi đè trong một trang

Ex3.1: AppTheme\Simple3\simple3.skin

<asp:Label BackColor="Orange" Font-Size="20pt" Runat="Server" />

<asp:TextBox BackColor="Yellow" BorderStyle="Dotted" Runat="Server" />

<asp:Button runat="server" ForeColor="LightYellow" BackColor="Red" Font-Size="12pt"/>

Ex3.2: 03ShowSkinTheme.aspx

<asp:Label ID="Label2" BackColor="Red" runat="server" Text="Họ và tên: "></asp:Label>

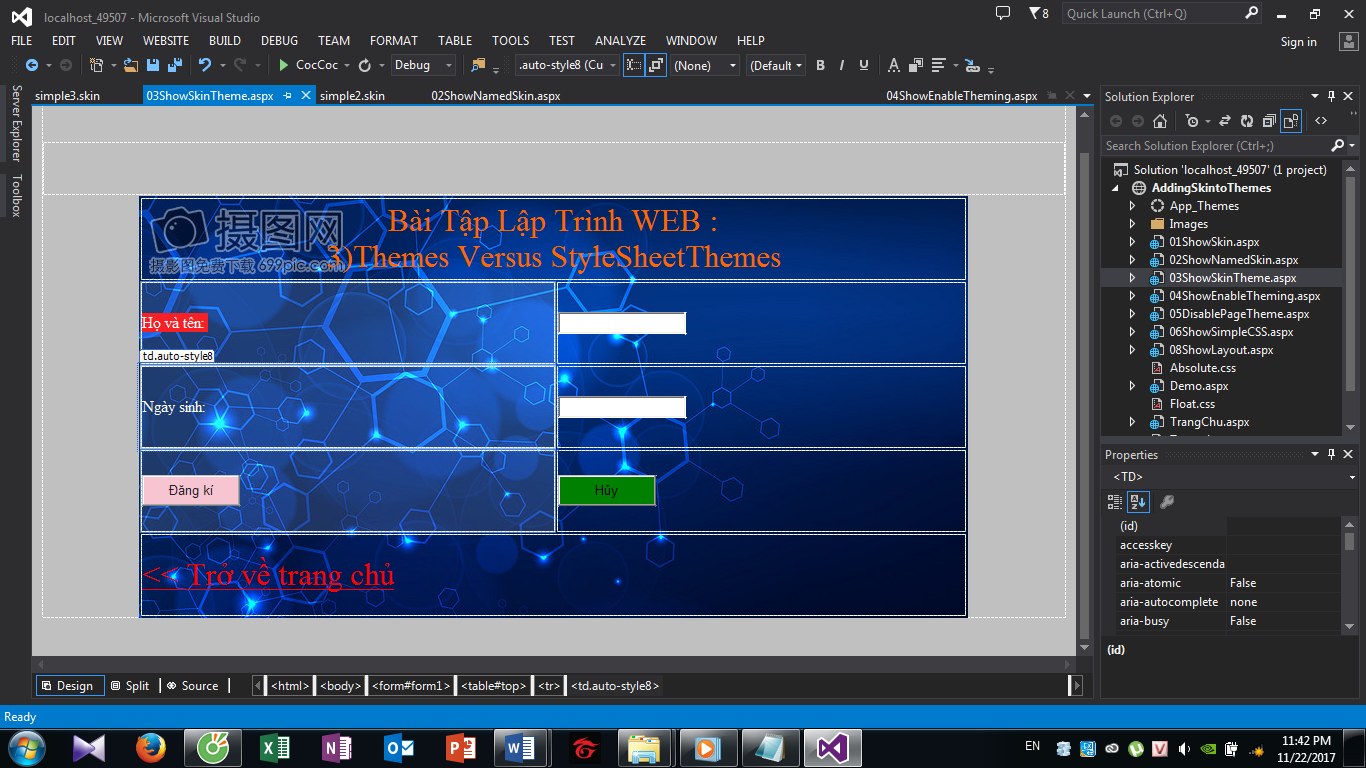
<asp:Label ID="Label3" runat="server" Text="Ngày sinh:"></asp:Label>

<asp:Button ID="Button2" BackColor="Pink" runat="server" Height="31px" Text="Đăng kí" Width="98px" />

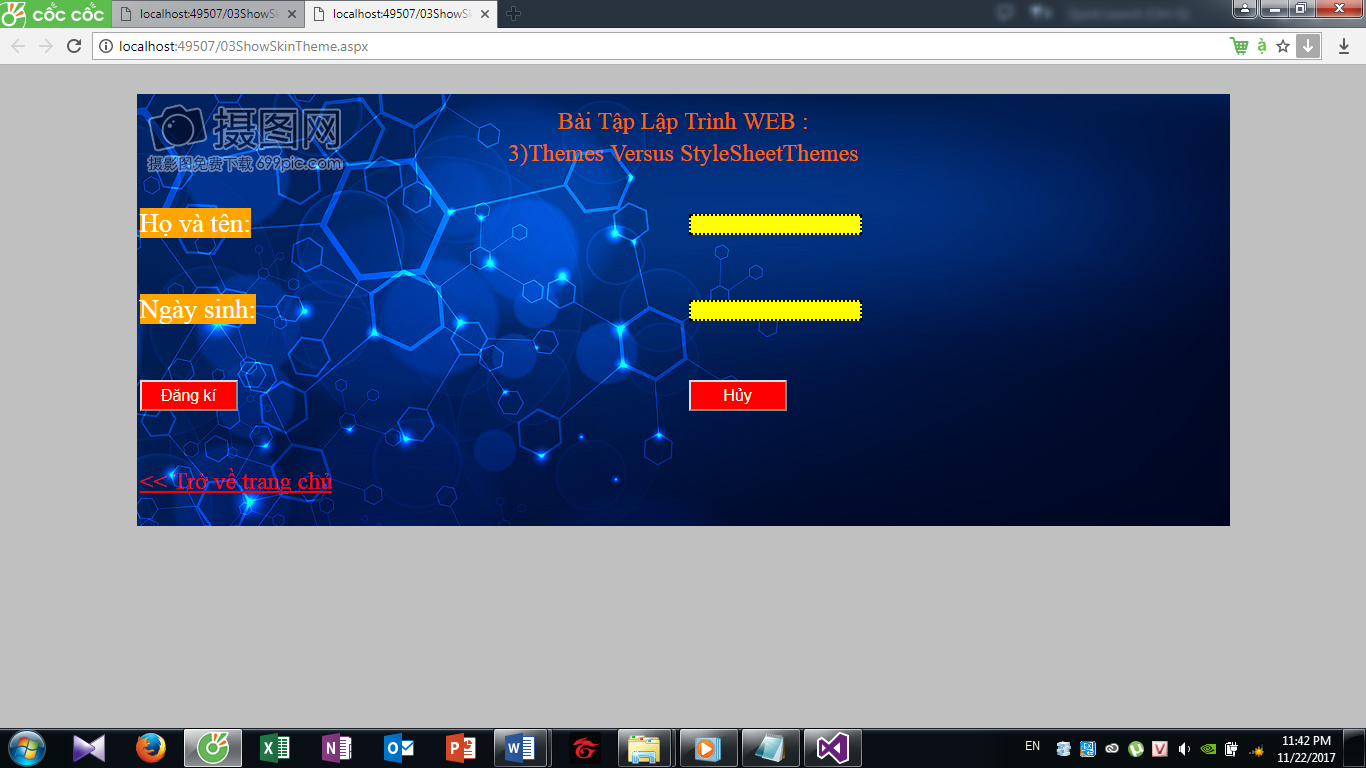
<asp:Button ID="Button1" BackColor="Green" runat="server" Height="31px" Text="Hủy" Width="98px" />

<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>

<asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>



Trang trong Ex3.2 bao gồm một Label có thuộc tính BackColor được thiết lập giá trị Red. Tuy nhiên, khi bạn mở trang, BackColor tuyên bố trong Skin ghi đè BackColor được khai báo trong trang, và Label hiển thị với background màu cam.



# **IV) Disabling Themes:**

Mỗi điều khiển ASP.NET bao gồm một thuộc tính EnableTheming. Bạn có thể sử dụng thuộc tính EnableTheming để ngăn không cho Skin được áp dụng cho một control cụ thể trong một trang.

Ví dụ, trang trong EX 4.1 chứa hai bộ điều khiển Lịch. Control lịch thứ hai có thuộc tính EnableTheming của nó được đặt thành giá trị False

Ex4.1: 04ShowEnableTheming.aspx

<asp:Calendar

id=”Calendar1”

Runat=”server” />

<br /><br />

<asp:Calendar

id=”Calendar2”

EnableTheming=”false”

Runat=”server” />

Trang trong Ex4.1 bao gồm một thuộc tính Theme áp dụng Simple Theme cho trang. Simple Theme bao gồm Skin trong liệt kê Ex4.2.

Ex4.2: Simple4\Calendar.skin

<asp:Calendar

BackColor=”White”

BorderColor=”White”

BorderWidth=”1px”

Font-Names=”Verdana”

Font-Size=”9pt”

ForeColor=”Black”

NextPrevFormat=”FullMonth”

Width=”400px”

Runat=”Server”>

<SelectedDayStyle

BackColor=”#333399”

ForeColor=”White” />

<OtherMonthDayStyle

ForeColor=”#999999” />

<TodayDayStyle

BackColor=”#CCCCCC” />

<NextPrevStyle

Font-Bold=”True”

Font-Size=”8pt”

ForeColor=”#333333”

VerticalAlign=”Bottom” />

<DayHeaderStyle

Font-Bold=”True”

Font-Size=”8pt” />

<TitleStyle

BackColor=”White”

BorderColor=”Black”

BorderWidth=”4px”

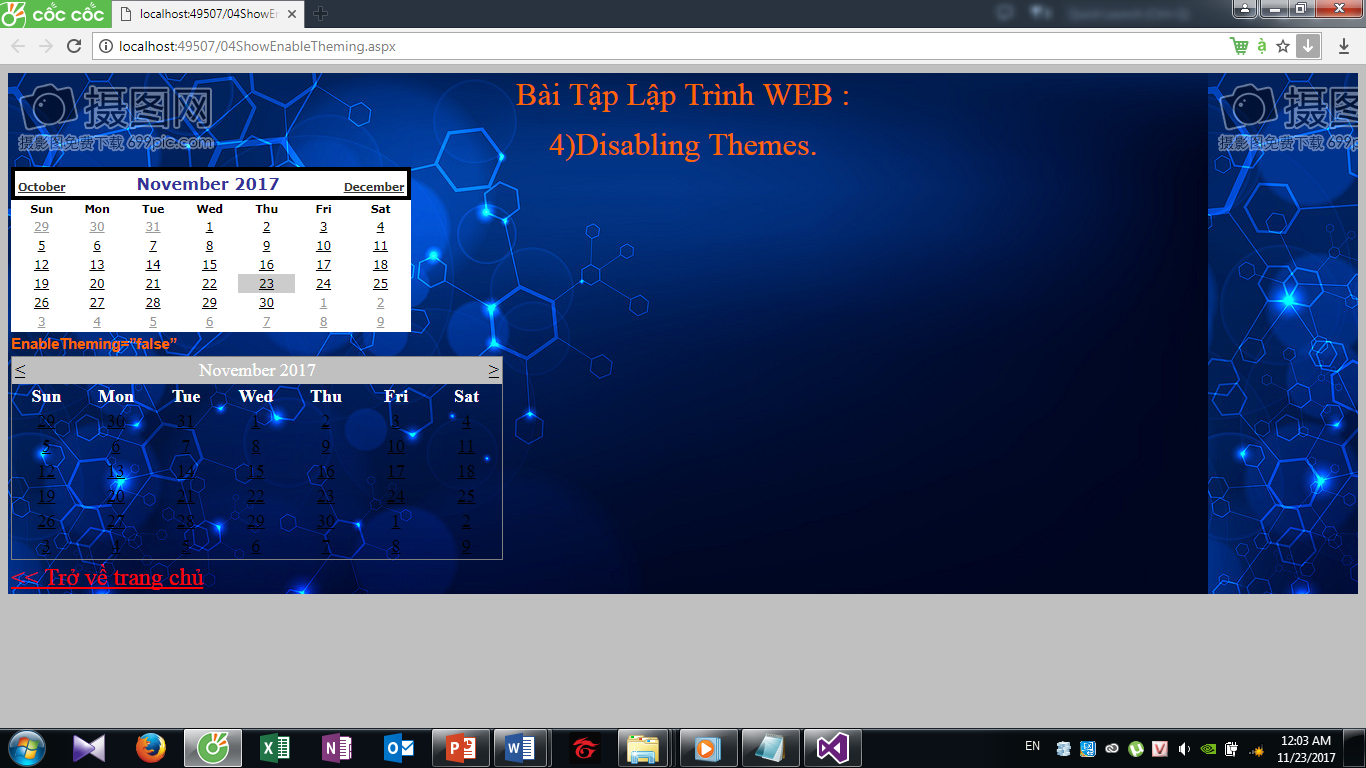
Font-Bold=”True”

Font-Size=”12pt”

ForeColor=”#333399” />

</asp:Calendar>

Khi bạn mở trang trong Ex4.2 trong một trình duyệt web, Skin được áp dụng cho trình duyệt điều khiển lịch đầu tiên nhưng không điều khiển lịch thứ hai.



# **V)Registering Themes in the Web Configuration File:**

Thay vì thêm thuộc tính Theme hoặc StyleSheetTheme cho mỗi trang và mỗi trang bạn đều muốn áp dụng một Theme, bạn có thể đăng ký Theme cho tất cả các trang trong ứng dụng của bạn trong tệp cấu hình web.

Ex5.1: Web.Config

<?xml version=”1.0”?>

<configuration>

<system.web>

<pages theme=”Simple3” />

</system.web>

</configuration>

Thay vì sử dụng thuộc tính theme, bạn có thể sử dụng thuộc tính stylesheetTheme để áp dụng Theme cho các trang trong một ứng dụng. Nếu bạn sử dụng thuộc tính styleSheetTheme, bạn có thể ghi đè đặc điểm Skin thuộc tính trong một trang.

Cấu hình web trong Ex 5.2 bao gồm thuộc tính stylesheetSheetTheme.

Ex 5.2 Web.Config

<?xml version=”1.0”?>

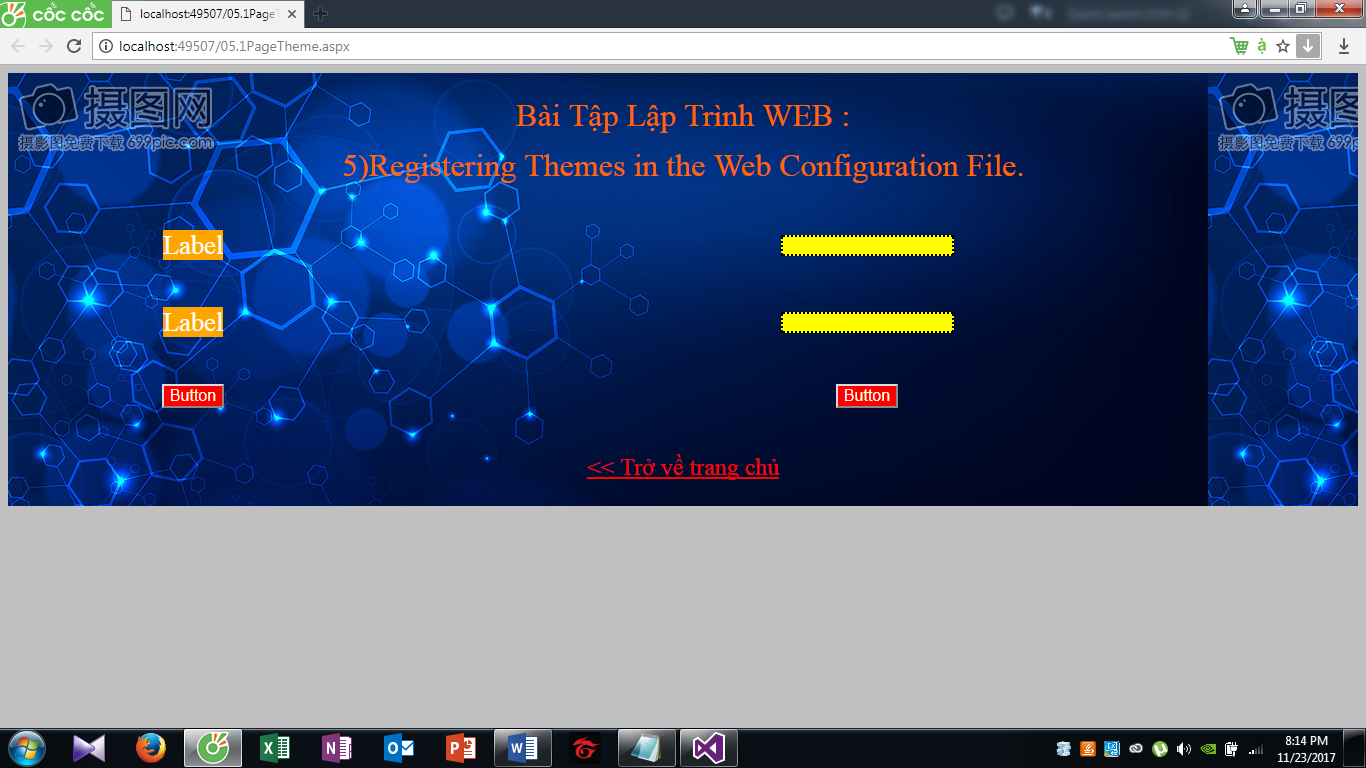
<configuration>

<system.web>

<pages styleSheetTheme=”Simple3” />

</system.web>

</configuration>



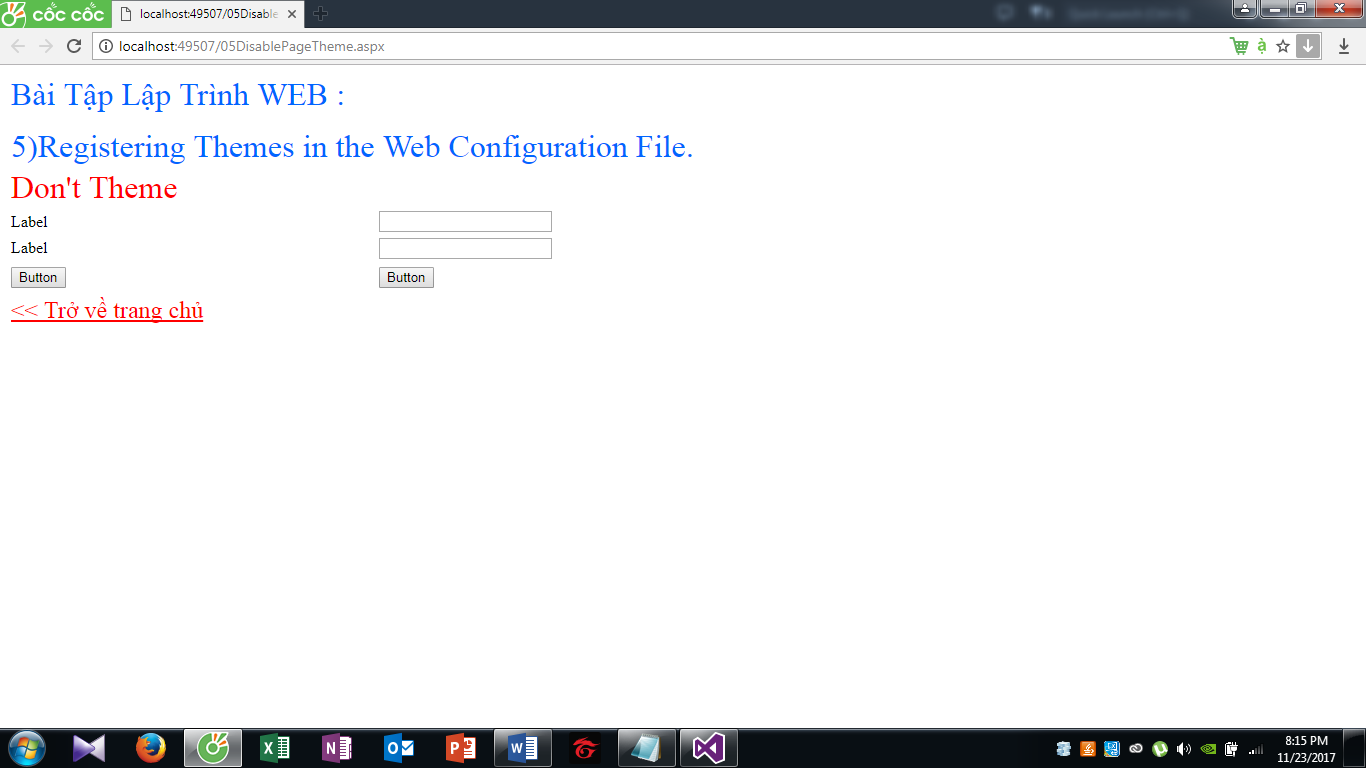
Sau khi bạn bật một chủ đề cho một ứng dụng, bạn có thể vô hiệu Theme cho một

bằng cách sử dụng thuộc tính EnableTheming với chỉ thị <% @ Page%>. Ví dụ,

trang trong Ex5.3 vô hiệu hóa bất kỳ Themes nào được cấu hình trong tệp cấu hình web.

Ex5.3 DisablePageTheme.aspx

<%@ Page Language=”C#” EnableTheming=”false” %>



# **VI)Adding Cascading Style Sheets to Themes:**

Là một thay thế cho Skins, bạn có thể sử dụng tệp CSS để điểu khiển sự xuất hiện của cả hai phần tử HTML và điều khiển ASP.NET chứa trong một trang. Nếu bạn thêm một tập tin CSS vào một Theme, CSS sẽ tự động được áp dụng cho mỗi trang mà Theme áp dụng.

Ex6.1 App\_Themes\StyleTheme\SimpleStyle.css

html

{

background-color:gray;

font:14px Georgia,Serif;

}

.content

{

margin:auto;

width:600px;

border:solid 1px black;

background-color:White;

padding:10px;

}

h1

{

color:Gray;

font-size:18px;

border-bottom:solid 1px orange;

}

label

{

font-weight:bold;

}

input

{

background-color:Yellow;

border:double 3px orange;

}

.button

{

background-color:#eeeeee;

}

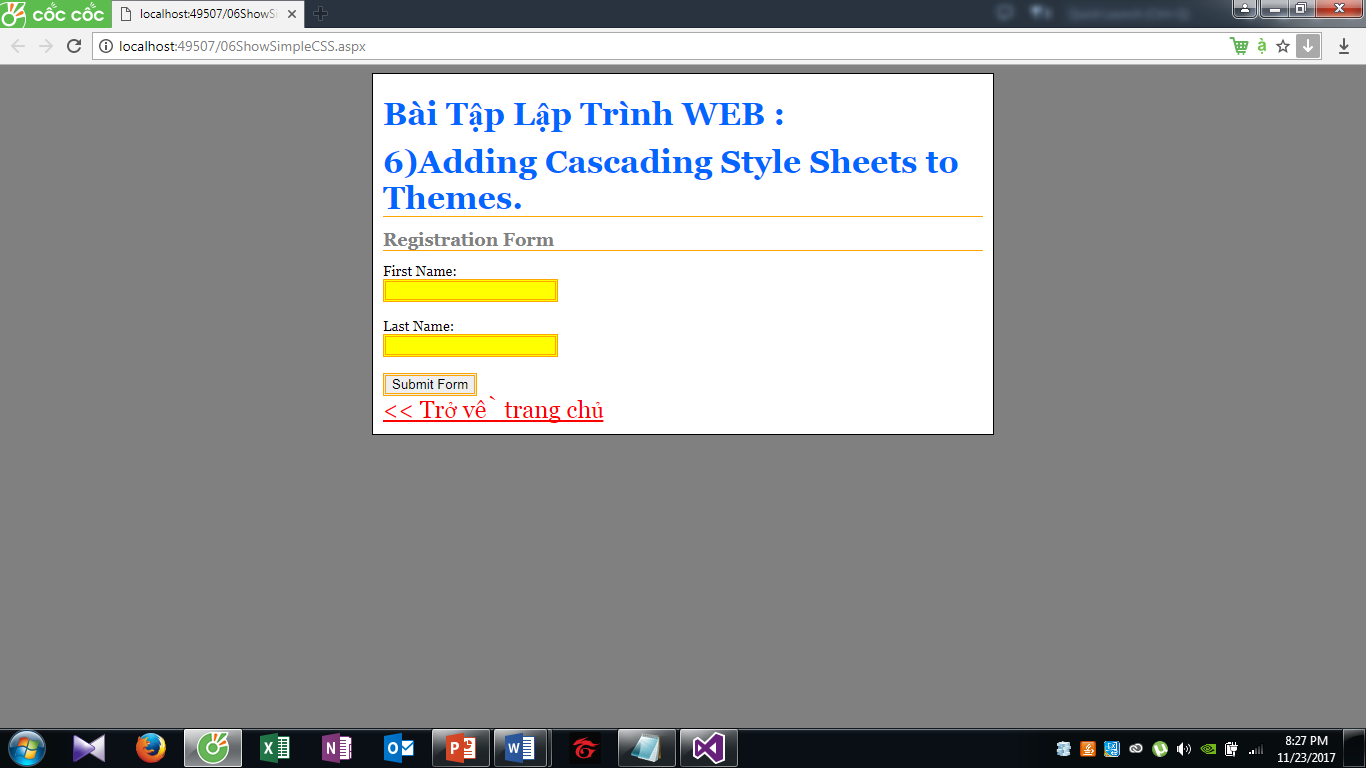
Nếu bạn thêm tệp SimpleStyle.css vào một Theme có tên StyleTheme (một thư mục có tên

StyleTheme trong thư mục App\_Themes), Cascading Style Sheet được áp dụng tự động

đến trong Ex6.2

Ex6.2 ShowSimpleCSS.aspx

<%@ Page Language=”C#” Theme=”StyleTheme” %>



Ưu điểm của việc sử dụng Cascading Style Sheets là chúng cho kết quả nhanh hơn khi tải trang. Nội dung nhiều hơn bạn có thể đặt trong một Style Sheet, thì càng ít nội dung phải được tải mỗi khi bạn thực hiện một yêu cầu trang. Nội dung của Style Sheet có thể được tải và lưu trữ bởi trình duyệt và được áp dụng cho tất cả các trang trong web ứng dụng

# **VII) Adding Multiple Cascading Style Sheets to a Theme:**

Bạn có thể thêm nhiều tệp CSS vào thư mục Theme khi cần thiết. Khi bạn thêm nhiều Cascading Style Sheets tới một chủ đề, tất cả các Cascading Style Sheets đều được áp dụng cho một trang khi Theme được áp dụng cho một trang.

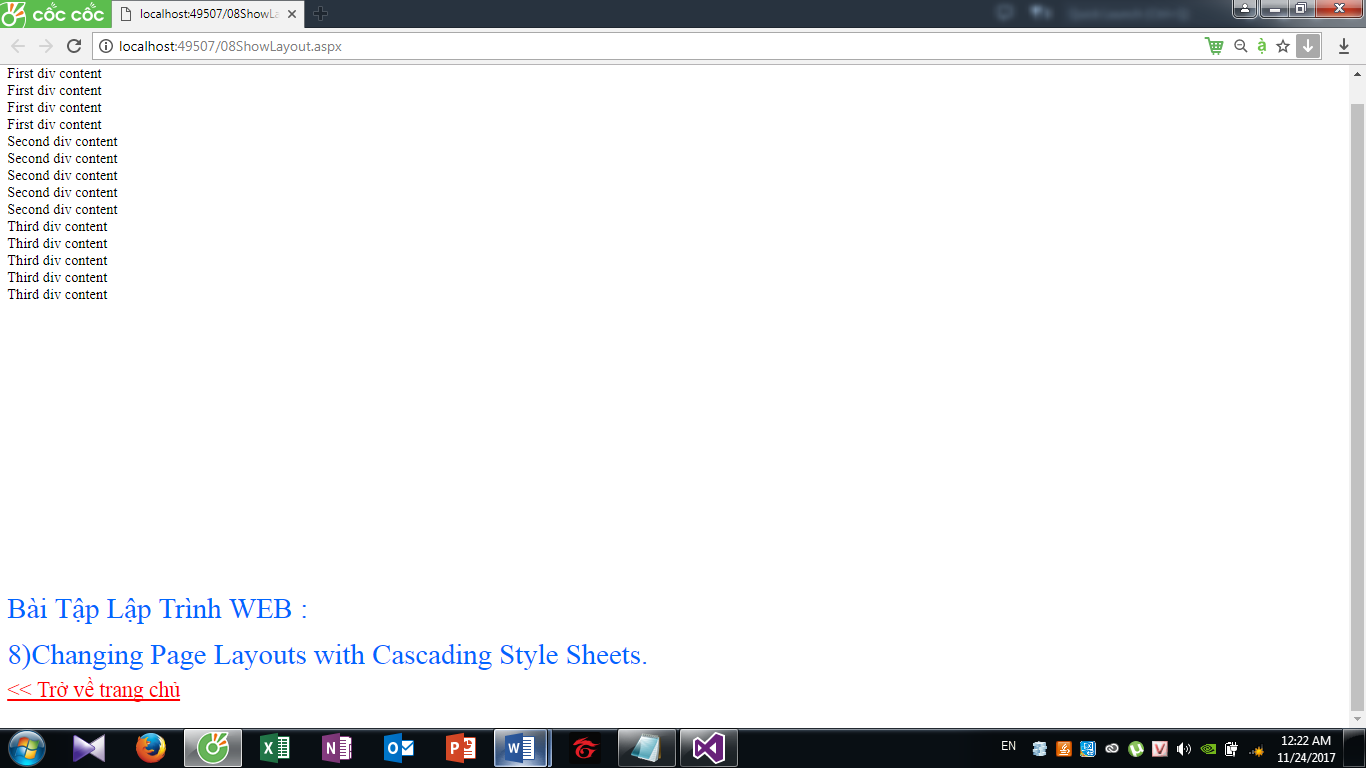


# **VIII)Changing Page Layouts with Cascading Style Sheets:**

Bởi vì bạn có thể sử dụng một Cascading Style Sheet để thay đổi cách bố trí của một trang, bạn có thể sử dụng một Theme để bố cục trang điều khiển.

Ví dụ, trang trong Ex8.1 chứa ba thẻ <div>. Theo mặc định, nếu bạn mở

trang, nội dung của các thẻ <div> được xếp chồng lên nhau.



Nếu bạn thêm Cascading Style Sheet trong Ex8.2, bạn có thể sửa đổi cách bố trí của

<div>

Ex8.2 Float.css

html

{

background-color:Silver;

font:14px Arial,Sans-Serif;

}

#div1

{

float:left;

width:25%;

margin:15px;

padding:10px;

background-color:White;

}

#div2

{

float:left;

width:25%;

margin:15px;

padding:10px;

background-color:White;

}

#div3

{

float:left;

width:25%;

margin:15px;

padding:10px;

background-color:White;

}

