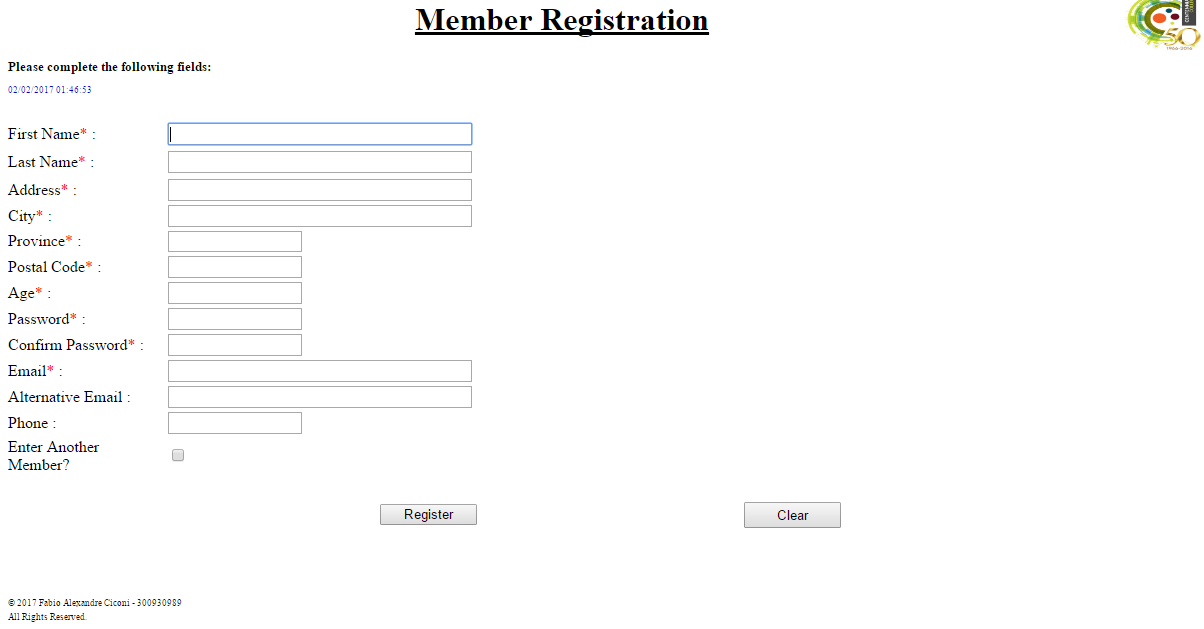
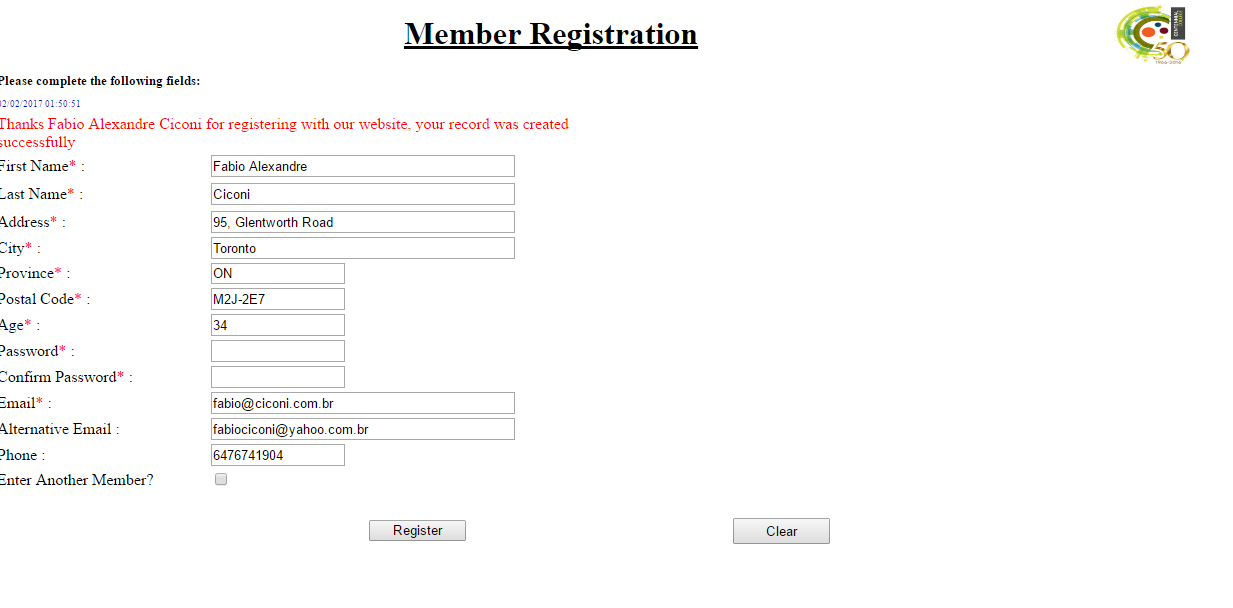
ASSIGNMENT #01

Fabio Alxandre Ciconi - 300930989

# Project Member Registration





# Summary

Contents

[Part I 2](#_Toc473922485)

[Part II 3](#_Toc473922486)

[Part I - Solutions 4](#_Toc473922487)

[Part II - Solutions 9](#_Toc473922488)

[Part II – Questions Part: 12](#_Toc473922489)

[HTML CODE 15](#_Toc473922490)

[ASP .NET AND C# 22](#_Toc473922491)

# Part I

1- Make sure that all the mandatory fields are already filled by the user when they want to register. (Fields with \*are mandatory fields) If there is any mandatory field that was left blank, the program should prompt the user to fill it.

2- The province should be exactly two characters e.g. ON, MN, SK, AB, BC and if the user entered them in small characters, the characters should be automatically converted to upper case.

3- The postal code should check if the user follows the a0a0a0 format, e.g. M6S4T1 (Hint: You need to search regular expression)

4- The program should check age, and if it is not between 18 to 120 the user should receive an appropriate message to correct the age content. (Hint: You need to search regular expression)

5- Passwords must have at least 6 characters and must contain at least one digit and one upper-case character. (Hint: You need to search regular expression)

6- The Confirm Password and Password fields should have identical input. If there is any difference, an appropriate message should inform the user.

7- The Email field must contain the @ and . characters. (Hint: You need to search regular expression)

8- As long as the “Enter another member” checkbox is true, the registration should be continued. (Hint: Using loop)

9- The register button should just save the last text contents into their variables, and the clear button should simply clear all the text boxes.

10- Simply use your common sense for anything that was not described. You can email me to query as well.

# Part II

1- When the form fields pass validation, an alert message is displayed to show a confirmation message (e.g. “Thanks for registering with our website, your record was created successfully.”).

2- Use one function to concatenate the name and the last name and show it in the success message, described in the above item. (item number 1)

3- You need to fully document your code, both in HTML and server side.

4- Change the website title to Centennial. (Changing the logo has a bonus point)

5- Store the content of each textbox into a variable. What kind of data types are you going to use for each of them? Why? e.g. I have chosen integer or string as the datatype for this field, and the reason is…

a. Suggestion: Use a table to ease your job, and explain briefly in that designed table in your word document that is explained as follow.

6- Provide a word document and explain the following items:

1. Briefly describe each segment of your code in your word document. You should be able to elaborate on the steps and what you have done through the code.
2. Use snapshots from your Visual Studio environment, your website and messages, and copy your entire code in the appendix section.

7- Explain the following questions in a separate section in your word document:

a. What kind of Directives did you use? Why?

b. Explain the role of View State in a few sentences.

c. What kind of logical expression did you use? Explain about it.

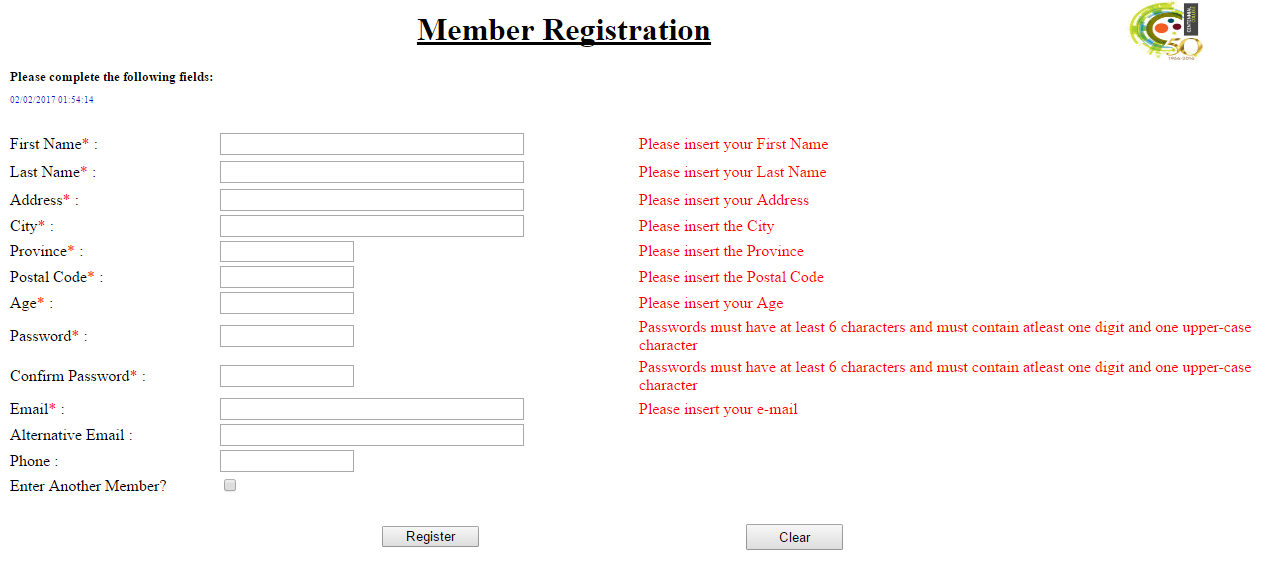
d. What kind of loop did you use? Why not the other type

e. What is the meaning of robustness? Did you use this feature in your project?

**Bonus Part for this assignment**

Upload the assignment document and supporting picture files to the student web server and create a hyperlink called Assignment01 from your home page to point to the assignment01file.

# Part I - Solutions

1. Make sure that all the mandatory fields are already filled by the user when they want to register. (Fields with \*are mandatory fields) If there is any mandatory field that was left blank, the program should prompt the user to fill it.
2. The province should be exactly two characters e.g. ON, MN, SK, AB, BC and if the user entered them in small characters, the characters should be automatically converted to upper case.

/// <summary>

/// Valid and check if it is a Canadian Province

/// </summary>

/// <param name="province\_aux"></param>

/// <returns></returns>

public bool Check\_Province\_Canada(string province\_aux)

{

if (province\_aux == "ON" || province\_aux == "QC" ||

province\_aux == "NS" || province\_aux == "NB" ||

province\_aux == "MB" || province\_aux == "BC" ||

province\_aux == "PE" || province\_aux == "SK" ||

province\_aux == "AB" || province\_aux == "NL")

{

return true;

}

return false;

}

1. -The postal code should check if the user follows the a0a0a0 format, e.g. M6S4T1 (Hint: You need to search regular expression).

///<summary>

///The method Check\_PostalCode, checks the postal code format and

///if it is Canadian

///<param name="TextPostalCode\_aux"></param>

///</summary>

public bool Check\_PostalCode(string TextPostalCode\_aux)

{

//Regular expression to valid Canadian Postal Code in the format of "M3A-1A5"

//Character Capital Letter: [ABCEGHJ-NPRSTVXY]

//Quantity{1}

//Number range [0-9]

//[-] separetor

string Pattern = "^[ABCEGHJ-NPRSTVXY]{1}[0-9]{1}[ABCEGHJ-NPRSTV-Z]{1}[-]?[0-9]{1}[ABCEGHJ-NPRSTV-Z]{1}[0-9]{1}$";

Regex reg = new Regex(Pattern, RegexOptions.IgnoreCase | RegexOptions.Compiled);

if (!(reg.IsMatch(TextPostalCode\_aux)))

{

return false;

}

return true;

}

4- The program should check age, and if it is not between 18 to 120 the user should receive an appropriate message to correct the age content. (Hint: You need to search regular expression)

/// <summary>

/// Consist if it is between 18 to 120

/// </summary>

/// <param name="Age\_Calc\_Aux"></param>

/// <returns></returns>

public bool Validade\_Age(int Age\_Calc\_Aux)

{

if ((Age\_Calc\_Aux > 18 && Age\_Calc\_Aux < 120))

{

//my test to see if it is ok

MyLabel\_Age.Text = "AGE OK";

return true;

}

else if (Age\_Calc\_Aux < 18)

{

MyLabel\_Age.Text = "OnLy over 18 years Old";

MyLabel\_Age.Visible = true;

}

else if (Age\_Calc\_Aux > 120)

{

MyLabel\_Age.Text = "YOUR ARE TOO OLD TO BE STUDENT.";

MyLabel\_Age.Visible = true;

}

return false;

}

1. Passwords must have at least 6 characters and must contain at least one digit and one upper-case character. (Hint: You need to search regular expression)

///<summary>

///The method Check\_Password, cheks the password format and

///check the first password is equal the confirmation's password

///<param name="Check\_Password\_Aux"></param>

///<param name="Check\_Password\_Aux\_Alt"></param>

///</summary>

public bool Check\_Password(string Check\_Password\_Aux, string Check\_Password\_Aux\_Alt)

{

//Regular expression to valid email

//At least one upper case english letter, (?=.\*?[A - Z])

//At least one lower case english letter, (?=.\*?[a - z])

//At least one digit, (?=.\*?[0 - 9])

//At least one special character, (?=.\*?[#?!@$%^&\*-])--- I do not use it in this example

//Minimum 6 in length.{6,} (with the anchors

string Pattern = "^(?=.\*?[A-Z])(?=.\*?[a-z])(?=.\*?[0-9]).{6,}$";

Regex reg = new Regex(Pattern, RegexOptions.IgnoreCase | RegexOptions.Compiled);

if (!(reg.IsMatch(Check\_Password\_Aux) && reg.IsMatch(Check\_Password\_Aux\_Alt)))

{

MyLabel\_Password.Text = "Passwords must have at least 6 characters and must contain at"+

"least one digit and one upper-case character";

MyLabel\_Password.Visible = true;

MyLabel\_PasswordConfirmation.Text = "Passwords must have at least 6 characters and must contain at" +

"least one digit and one upper-case character";

MyLabel\_PasswordConfirmation.Visible = true;

return false;

}

6- The Confirm Password and Password fields should have identical input. If there is any difference, an appropriate message should inform the user.

if (Check\_Password\_Aux != Check\_Password\_Aux\_Alt)

{

MyLabel\_Password.Text = "ATENTION - Password doens't match";

MyLabel\_Password.Visible = true;

MyLabel\_PasswordConfirmation.Text = "ATENTION - Password doens't match";

MyLabel\_PasswordConfirmation.Visible = true;

return false;

}

7- The Email field must contain the @ and . characters. (Hint: You need to search regular expression)

/// <summary>

/// Check if it is a valid email

/// </summary>

/// <param name="email\_aux"></param>

/// <returns></returns>

public bool Check\_Email(string email\_aux)

{

string Email\_Patern = @"^([a-zA-Z0-9\_\-\.]+)@((\[[0-9]{1,3}" +

@"\.[0-9]{1,3}\.[0-9]{1,3}\.)|(([a-zA-Z0-9\-]+\" +

@".)+))([a-zA-Z]{2,4}|[0-9]{1,3})(\]?)$";

Regex reg = new Regex(Email\_Patern, RegexOptions.IgnoreCase | RegexOptions.Compiled);

if (!(reg.IsMatch(email\_aux)))

{

return false;

}

return true;

}

8- As long as the “Enter another member” checkbox is true, the registration should be continued. (Hint: Using loop)

if (Validation\_ok == true)

{

//Show the success message

MyLabel\_MenssageResult.Visible = true;

string Student\_Full\_Name = string.Concat(TextFirstName.Text," ", TextLastName.Text);

MyLabel\_MenssageResult.Text = "Thanks " + Student\_Full\_Name + " for registering with our website, your record was created successfully";

while (MyCheck\_AnotherMember.Checked == true && Validation\_ok == true)

{

Clean\_TextBoxes();

lblMensagem\_Quant\_Regs.Text = "Please register a new student";

MyLabel\_MenssageResult.Visible = false;

MyCheck\_AnotherMember.Checked = false;

9- The register button should just save the last text contents into their variables, and the clear button should simply clear all the text boxes.

string Last\_Reg\_FirstName = TextFirstName.Text;

string Last\_Reg\_LastName = TextLastName.Text;

string Last\_Reg\_TextAddress = TextAddress.Text;

string Last\_Reg\_TextCity = TextCity.Text;

string Last\_Reg\_TextProvince = TextProvince.Text ;

string Last\_Reg\_TextPostalCode = TextPostalCode.Text ;

string Last\_Reg\_TextAge = TextAge.Text;

string Last\_Reg\_TextPassword = TextPassword.Text ;

string Last\_Reg\_TextPasswordConfirmation = TextPasswordConfirmation.Text;

string Last\_Reg\_TextEmail = TextEmail.Text;

string Last\_Reg\_TextEmailAlternative = TextEmail\_Alternative.Text;

string Last\_Reg\_TextPhone = TextPhone.Text;

}

}

}

# Part II - Solutions

1. When the form fields pass validation, an alert message is displayed to show a confirmation message (e.g. “Thanks for registering with our website, your record was created successfully.”).

if (Validation\_ok == true)

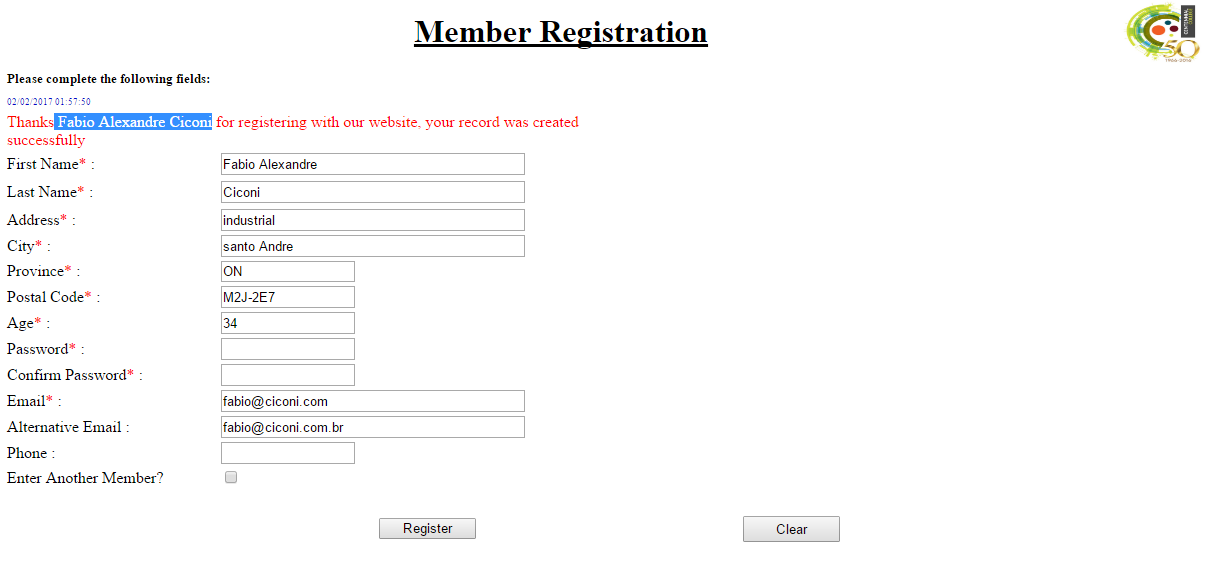
{

//Show the success message

MyLabel\_MenssageResult.Visible = true;

string Student\_Full\_Name = string.Concat(TextFirstName.Text," ", TextLastName.Text);

MyLabel\_MenssageResult.Text = "Thanks " + Student\_Full\_Name + " for registering with our website, your record was created successfully";

1.  Use one function to concatenate the name and the last name and show it in the success message, described in the above item. (item number 1)
2. You need to fully document your code, both in HTML and server side.



4- Change the website title to Centennial. (Changing the logo has a bonus point)

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<!-- Web site Title -->

<title>Centennial</title>

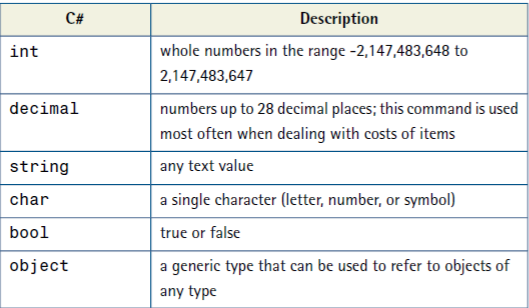
<style type="text/css">

.newStyle1 {

font-family: Arial;

text-decoration: underline;

}

5- Store the content of each textbox into a variable. What kind of data types are you going to use for each of them? Why? e.g. I have chosen integer or string as the datatype for this field, and the reason is…

1. Suggestion: Use a table to ease your job, and explain briefly in that designed table in your word document that is explained as follow.

In my project, I, have chosen int to store the content of Age, due to necessity to make calculations. However, to the name, last name, address, city province, postal code, password, password confirmation, email, email alternative and phone I’ve choose string because all fields can be fill in either with numbers or characters. Finally, to field “enter another member”, I have used bool because the behavior of this fields, when it is check is true, otherwise it is false.

6- Provide a word document and explain the following items:

1. Briefly describe each segment of your code in your word document. You should be able to elaborate on the steps and what you have done through the code.

My code start showing the actual date and time int load page method. After, the user fill out the fields and press register, I started validate each field whether is correct or not.

1. Use snapshots from your Visual Studio environment, your website and messages, and copy your entire code in the appendix section.

//done//

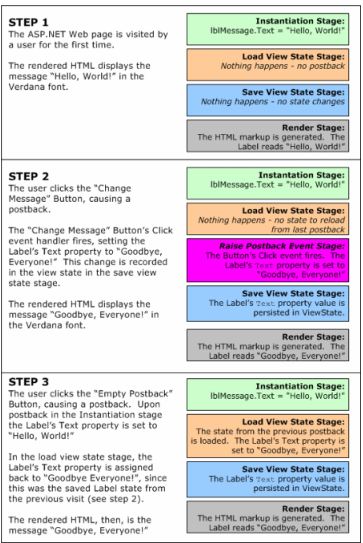
# Part II – Questions Part:

7- Explain the following questions in a separate section in your word document:

a. What kind of Directives did you use? Why?

|  |  |  |
| --- | --- | --- |
| **Directive** | **Description** | **Why?** |
| [@ Assembly](https://msdn.microsoft.com/en-us/library/d864zc1k.aspx) | Links an assembly to the current page or user control declaratively. | No used it. |
| [@ Control](https://msdn.microsoft.com/en-us/library/d19c0t4b.aspx) | Defines control-specific attributes used by the ASP.NET page parser and compiler and can be included only in .ascx files (user controls). | No used it. |
| [@ Implements](https://msdn.microsoft.com/en-us/library/cbsf6k72.aspx) | Indicates that a page or user control implements a specified .NET Framework interface declaratively. | No used it. |
| [@ Import](https://msdn.microsoft.com/en-us/library/eb44kack.aspx) | Imports a namespace into a page or user control explicitly. | No used it. |
| [@ Master](https://msdn.microsoft.com/en-us/library/ms228176.aspx) | Identifies a page as a master page and defines attributes used by the ASP.NET page parser and compiler and can be included only in .master files. | No used it. |
| [@ MasterType](https://msdn.microsoft.com/en-us/library/ms228274.aspx) | Defines the class or virtual path used to type the [Master](https://msdn.microsoft.com/en-us/library/system.web.ui.page.master.aspx) property of a page. | No used it. |
| [@ OutputCache](https://msdn.microsoft.com/en-us/library/hdxfb6cy.aspx) | Controls the output caching policies of a page or user control declaratively. | No used it. |
| [@ Page](https://msdn.microsoft.com/en-us/library/ydy4x04a.aspx) | Defines page-specific attributes used by the ASP.NET page parser and compiler and can be included only in .aspx files. | Language: I used to define the code language(c#).  AutoEventWireup: Indicates whether the page's events are autowired.  CodeBehind: to link .cs file;  Inherits: to defines a code-behind class for the page to inherit |
| [@ PreviousPageType](https://msdn.microsoft.com/en-us/library/ms228169.aspx) | Creates a strongly typed reference to the source page from the target of a cross-page posting. | No used it. |
| [@ Reference](https://msdn.microsoft.com/en-us/library/w70c655a.aspx) | Links a page, user control, or COM control to the current page or user control declaratively. | No used it. |
| [@ Register](https://msdn.microsoft.com/en-us/library/c76dd5k1.aspx) | Associates aliases with namespaces and classes, which allow user controls and custom server controls to be rendered when included in a requested page or user control. | No used it. |

1. Explain the role of View State in a few sentences.

It a technique that Asp .net use to preserve page and control values between postbacks.

c. What kind of logical expression did you use? Explain about it.

I used logical expression such as == to make comparison

!= to check if the fields are different between them,

|| to check, for example the province and

&& to put more than one validation.

d. What kind of loop did you use? Why not the other type

I used a while loop to insert new register when the checkbox is checked. In that part I needed a simple loop to compere 2 variables. Other type of loop such as case does not fit well, and it would be to complicate, I didn’t need an incremental control and I don’t know how many register will be included by the user.

1. What is the meaning of robustness? Did you use this feature in your project?

It is a way to coding that prevents abnormal termination or unexpected actions  
I've tried to make it as robust as possible with what I've learned so far.

# HTML CODE

<html xmlns="http://www.w3.org/1999/xhtml">

<head runat="server">

<!-- Web site Title -->

<title>Centennial</title>

<style type="text/css">

.newStyle1 {

font-family: Arial;

text-decoration: underline;

}

.auto-style1 {

color: #FF0000;

}

.auto-style12 {

width: 16%;

}

.auto-style14 {

font-size: xx-small;

}

.auto-style15 {

width: 320px;

}

.auto-style16 {

width: 89px;

}

.auto-style17 {

width: 74px;

}

.auto-style19 {

width: 74px;

text-align: center;

}

.auto-style20 {

width: 89px;

text-align: center;

}

.auto-style23 {

font-size: small;

}

.auto-style24 {

width: 52%;

}

.newStyle2 {

}

.auto-style25 {

width: 369px;

}

.auto-style26 {

width: 437px;

font-size: xx-small;

}

.auto-style27 {

width: 74px;

font-size: xx-small;

}

.auto-style28 {

width: 89px;

font-size: xx-small;

}

.auto-style29 {

width: 1108px;

text-align: center;

text-decoration: underline;

font-size: xx-large;

}

.auto-style32 {

font-size: xx-large;

}

.auto-style33 {

width: 181px;

}

.auto-style34 {

width: 320px;

font-size: xx-small;

}

.auto-style36 {

width: 70px;

text-align: center;

text-decoration: underline;

font-size: xx-large;

}

.auto-style37 {

width: 16%;

height: 26px;

}

.auto-style38 {

width: 181px;

height: 26px;

}

.auto-style39 {

width: 52%;

height: 26px;

}

</style>

</head>

<body>

<form id="Principalform" runat="server">

<!-- Title Page and Image logo-->

<table class="Head">

<tr>

<td class="auto-style29">

<strong>Member Registration</strong></td>

<td class="auto-style36">

<strong><asp:Image ID="ImageLogo" runat="server" Height="63px" ImageAlign="Right"

ImageUrl="~/Images/MItwiXGq.jpg" Width="87px" style="margin-left: 0px" CssClass="auto-style32" />

</strong>

</td>

</tr>

</table>

<!-- Body Part -->

<table class="Body">

<tr>

<!-- This label shows a warning to fill in the fields-->

<td class="Information-Row" colspan="3">

<strong><span class="auto-style23">Please complete the following fields:&nbsp;&nbsp;</span></strong></td>

</tr>

<tr>

<!--Shows actual date and time-->

<td class="Date-Time">

<asp:Label ID="myTimeLabel" runat="server" style="color: #0000CC; font-size: xx-small"></asp:Label>

</td>

<td class="auto-style33">&nbsp;</td>

<td class="auto-style24">&nbsp;</td>

</tr>

<tr>

<!--Show Warning whether the register was succesfully inserted or not-->

<td class="Warning" colspan="2">

<asp:Label ID="MyLabel\_MenssageResult" runat="server" style="color: #FF0000"></asp:Label>

</td>

<td class="auto-style24">&nbsp;</td>

</tr>

<tr>

<!-- Text box to insert the First Name's student-->

<td class="auto-style12">First Name<span class="auto-style1">\*</span> : </td>

<td class="auto-style33">

<asp:TextBox ID="TextFirstName" runat="server" Width="300px" OnTextChanged="TextFirstName\_TextChanged" Height="16px"></asp:TextBox>

</td>

<!-- Label First Name's student-->

<td class="auto-style24">

<asp:Label ID="MyLabel\_First" runat="server" CssClass="auto-style1"></asp:Label>

</td>

</tr>

<tr>

<!-- Text box to insert the Last Name's student-->

<td class="auto-style37">Last Name<span class="auto-style1">\*</span> : </td>

<td class="auto-style38">

<asp:TextBox ID="TextLastName" runat="server" Width="300px" height="16px"></asp:TextBox>

</td>

<!-- Label Last Name's student-->

<td class="auto-style39">

<asp:Label ID="MyLabel\_Last" runat="server" CssClass="auto-style1"></asp:Label>

</td>

</tr>

<tr>

<!-- Text box to insert the Address-->

<td class="auto-style12">Address<span class="auto-style1">\*</span> :</td>

<td class="auto-style33">

<asp:TextBox ID="TextAddress" runat="server"

style="margin-left: 0px" Width="300px" height="16px" OnTextChanged="TextAddress\_TextChanged"></asp:TextBox>

</td>

<!-- Label Address-->

<td class="auto-style24">

<asp:Label ID="MyLabel\_Address" runat="server" CssClass="auto-style1"></asp:Label>

</td>

</tr>

<tr>

<!-- Text box to insert the City Name's student-->

<td class="auto-style12">City<span class="auto-style1">\*</span> :&nbsp; </td>

<td class="auto-style33">

<asp:TextBox ID="TextCity" runat="server" Width="300px" height="16px"></asp:TextBox>

</td>

<!-- Label City Name-->

<td class="auto-style24">

<asp:Label ID="MyLabel\_City" runat="server" CssClass="auto-style1"></asp:Label>

</td>

</tr>

<tr>

<!-- Text box to insert the Province-->

<td class="auto-style12">Province<span class="auto-style1">\*</span> :&nbsp; </td>

<td class="auto-style33">

<asp:TextBox ID="TextProvince" runat="server" CausesValidation="True" Width="130px"

OnTextChanged="TextProvince\_TextChanged" MaxLength="2" ></asp:TextBox>

</td>

<!-- Label Province-->

<td class="auto-style24"><asp:Label ID="MyLabel\_Province" runat="server" CssClass="auto-style1"></asp:Label>

</td>

</tr>

<tr>

<!-- Text box to insert the Postal code-->

<td class="auto-style12">Postal Code<span class="auto-style1">\*</span> : </td>

<td class="auto-style33">

<asp:TextBox ID="TextPostalCode" runat="server" CausesValidation="True"

MaxLength="7" Width="130px" height="16px" ToolTip="Post Code a0a0a0 format, e.g. M6S4T1 "></asp:TextBox>

</td>

<!-- Postal Code Label-->

<td class="auto-style24">

<asp:Label ID="MyLabel\_Postal" runat="server" CssClass="auto-style1"></asp:Label>

</td>

</tr>

<tr>

<!-- Text box to insert the age-->

<td class="auto-style12">Age<span class="auto-style1">\*</span> :&nbsp; </td>

<td class="auto-style33">

<asp:TextBox ID="TextAge" runat="server" TextMode="Number" Width="130px"

height="16px" style="margin-left: 0px"></asp:TextBox>

</td>

<!-- Label Age -->

<td class="auto-style24">

<asp:Label ID="MyLabel\_Age" runat="server" CssClass="auto-style1"></asp:Label>

</td>

</tr>

<tr>

<!-- Text box to insert the Password -->

<td class="auto-style12">Password<span class="auto-style1">\*</span> :&nbsp;&nbsp;&nbsp;</td>

<td class="auto-style33">

<asp:TextBox ID="TextPassword" runat="server" CausesValidation="True" MaxLength="6"

TextMode="Password" ToolTip="Only 6 Characters" Width="130px" height="16px" style="margin-left: 0px"></asp:TextBox>

</td>

<!-- Label Password -->

<td class="auto-style24">

<asp:Label ID="MyLabel\_Password" runat="server" CssClass="auto-style1"></asp:Label>

</td>

</tr>

<tr>

<!-- Text box to insert the password again -->

<td class="auto-style12">Confirm Password<span class="auto-style1">\*</span> : </td>

<td class="auto-style33">

<asp:TextBox ID="TextPasswordConfirmation" runat="server" CausesValidation="True" MaxLength="6"

TextMode="Password" ToolTip="Only 6 Character" Width="130px" height="16px"></asp:TextBox>

</td>

<!-- Password Confirmation-->

<td class="auto-style24">

<asp:Label ID="MyLabel\_PasswordConfirmation" runat="server" CssClass="auto-style1"></asp:Label>

</td>

</tr>

<tr>

<!-- Text box to insert the email -->

<td class="auto-style12">Email<span class="auto-style1">\*</span> : </td>

<td class="auto-style33">

<asp:TextBox ID="TextEmail" runat="server" Width="300px" height="16px" style="margin-left: 0px"></asp:TextBox>

</td>

<!-- label Email -->

<td class="auto-style24">

<asp:Label ID="MyLabel\_Email" runat="server" CssClass="auto-style1"></asp:Label>

</td>

</tr>

<tr>

<!-- Text box to insert the alternative email -->

<td class="auto-style12">Alternative Email : </td>

<td class="auto-style33">

<asp:TextBox ID="TextEmail\_Alternative" runat="server" Width="300px" height="16px"></asp:TextBox>

</td>

<!-- Label Email alternative-->

<td class="auto-style24">

<asp:Label ID="MyLabel\_EmailAlt" runat="server" CssClass="auto-style1"></asp:Label>

</td>

</tr>

<tr>

<!-- Text box to insert Phone number -->

<td class="auto-style12">Phone : </td>

<td class="auto-style33">

<asp:TextBox ID="TextPhone" runat="server" TextMode="Phone"

Width="130px" height="16px" OnTextChanged="TextPhone\_TextChanged"></asp:TextBox>

</td>

<td class="auto-style24">&nbsp;</td>

</tr>

<tr>

<!-- Checkbox to insert a new register-->

<td class="auto-style12">Enter Another Member?</td>

<td class="auto-style33"><asp:CheckBox ID="MyCheck\_AnotherMember" runat="server" OnCheckedChanged="CheckAnotherMember\_CheckedChanged" />

</td>

<td class="auto-style24">&nbsp;</td>

</tr>

<tr>

<td class="auto-style12">&nbsp;</td>

<td class="auto-style33">&nbsp;</td>

<td class="auto-style24">&nbsp;</td>

</tr>

</table>

<!-- Botton Part-->

<table class="Footer">

<tr>

<!-- Show the massage to user if the register was inserted -->

<td class="auto-style25">

<asp:Label ID="lblMensagem\_Quant\_Regs" runat="server"></asp:Label>

</td>

<!-- Button Processar - Register the new user -->

<td class="auto-style20">

<asp:Button ID="ButtonProcessar" runat="server" Text="Register" onclick="ButtonProcessar\_Click"

style="text-align: center" Width="97px" />

</td>

<!-- Button Clear - Clear all register on the actual page -->

<td class="auto-style16">&nbsp;</td>

<td class="auto-style19">

<asp:Button ID="ButtonClear" runat="server" Text="Clear" onclick="ButtonClear\_Click" Width="97px" style="top: 598px; left: 567px; height: 26px" />

</td>

<td class="auto-style15">&nbsp;</td>

</tr>

<tr>

<td class="auto-style25">&nbsp;</td>

<td class="auto-style16">&nbsp;</td>

<td class="auto-style16">&nbsp;</td>

<td class="auto-style17">&nbsp;</td>

<td class="auto-style15">&nbsp;</td>

</tr>

<tr>

<td class="auto-style25">&nbsp;</td>

<td class="auto-style16">&nbsp;</td>

<td class="auto-style16">&nbsp;</td>

<td class="auto-style17">&nbsp;</td>

<td class="auto-style15">&nbsp;</td>

</tr>

<tr>

<td class="auto-style25">&nbsp;</td>

<td class="auto-style16">&nbsp;</td>

<td class="auto-style16">&nbsp;</td>

<td class="auto-style17">&nbsp;</td>

<td class="auto-style15">&nbsp;</td>

</tr>

<tr>

<!-- Copyright/ made by Fabio -->

<td class="auto-style14" colspan="2">© 2017 Fabio Alexandre Ciconi - 300930989 </td>

<td class="auto-style27">&nbsp;</td>

<td class="auto-style26">&nbsp;</td>

</tr>

<tr>

<!-- Label all rights reserved-->

<td class="auto-style14" colspan="2">All Rights Reserved.</td>

<td class="auto-style28">&nbsp;</td>

<td class="auto-style27">&nbsp;</td>

<td class="auto-style34">&nbsp;</td>

</tr>

</table>

</form>

</body>

</html>

# ASP .NET AND C#

<script runat="server">

/// <summary>

/// Page Load - show date an hour when the page is load

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

protected void Page\_Load(object sender, EventArgs e)

{

//Show current date and time to the user

myTimeLabel.Text = DateTime.Now.ToString();

}

//

/// <summary>

/// Clear all textboxes

/// </summary>

public void Clean\_TextBoxes()

{

TextFirstName.Text = string.Empty;

TextLastName.Text = string.Empty;

TextAddress.Text = string.Empty;

TextCity.Text = string.Empty;

TextProvince.Text = string.Empty;

TextPostalCode.Text = string.Empty;

TextAge.Text = string.Empty;

TextPassword.Text = string.Empty;

TextPasswordConfirmation.Text = string.Empty;

TextEmail.Text = string.Empty;

TextEmail\_Alternative.Text = string.Empty;

TextPhone.Text = string.Empty;

}

/// <summary>

/// Valid and check if it is a Canadian Province

/// </summary>

/// <param name="province\_aux"></param>

/// <returns></returns>

public bool Check\_Province\_Canada(string province\_aux)

{

if (province\_aux == "ON" || province\_aux == "QC" ||

province\_aux == "NS" || province\_aux == "NB" ||

province\_aux == "MB" || province\_aux == "BC" ||

province\_aux == "PE" || province\_aux == "SK" ||

province\_aux == "AB" || province\_aux == "NL")

{

return true;

}

return false;

}

/// <summary>

/// Check if it is a valid email

/// </summary>

/// <param name="email\_aux"></param>

/// <returns></returns>

public bool Check\_Email(string email\_aux)

{

string Email\_Patern = @"^([a-zA-Z0-9\_\-\.]+)@((\[[0-9]{1,3}" +

@"\.[0-9]{1,3}\.[0-9]{1,3}\.)|(([a-zA-Z0-9\-]+\" +

@".)+))([a-zA-Z]{2,4}|[0-9]{1,3})(\]?)$";

Regex reg = new Regex(Email\_Patern, RegexOptions.IgnoreCase | RegexOptions.Compiled);

if (!(reg.IsMatch(email\_aux)))

{

return false;

}

return true;

}

/// <summary>

/// Consist if it is between 18 to 120

/// </summary>

/// <param name="Age\_Calc\_Aux"></param>

/// <returns></returns>

public bool Validade\_Age(int Age\_Calc\_Aux)

{

if ((Age\_Calc\_Aux > 18 && Age\_Calc\_Aux < 120))

{

//my test to see if it is ok

MyLabel\_Age.Text = "AGE OK";

return true;

}

else if (Age\_Calc\_Aux < 18)

{

MyLabel\_Age.Text = "OnLy over 18 years Old";

MyLabel\_Age.Visible = true;

}

else if (Age\_Calc\_Aux > 120)

{

MyLabel\_Age.Text = "YOUR ARE TOO OLD TO BE STUDENT.";

MyLabel\_Age.Visible = true;

}

return false;

}

/// <summary>

/// Button Clear- Clear text boxes and all labels

/// </summary>

/// <param name="s"></param>

/// <param name="e"></param>

public void ButtonClear\_Click(Object s, EventArgs e)

{

//delete all textboxes

TextFirstName.Text = string.Empty;

TextLastName.Text = string.Empty;

TextAddress.Text = string.Empty;

TextCity.Text = string.Empty;

TextProvince.Text = string.Empty;

TextPostalCode.Text = string.Empty;

TextAge.Text = string.Empty;

TextPassword.Text = string.Empty;

TextPasswordConfirmation.Text = string.Empty;

TextEmail.Text = string.Empty;

TextEmail\_Alternative.Text = string.Empty;

TextPhone.Text = string.Empty;

//labels become invisible

MyLabel\_First.Visible = false;

MyLabel\_Last.Visible = false;

MyLabel\_Age.Visible = false;

MyLabel\_Address.Visible = false;

MyLabel\_City.Visible = false;

MyLabel\_Password.Visible = false;

MyLabel\_PasswordConfirmation.Visible = false;

MyLabel\_Postal.Visible = false;

MyLabel\_Province.Visible = false;

MyLabel\_Email.Visible = false;

MyLabel\_MenssageResult.Visible = false;

MyLabel\_EmailAlt.Visible = false;

}

/// <summary>

/// Consist mandatory fields

/// </summary>

/// <param name="name\_men\_aux"></param>

/// <param name="last\_men\_aux"></param>

/// <param name="address\_men\_aux"></param>

/// <param name="city\_men\_aux"></param>

/// <param name="province\_men\_aux"></param>

/// <param name="postal\_men\_aux"></param>

/// <param name="age\_men\_aux"></param>

/// <param name="pass1\_men\_aux"></param>

/// <param name="pass2\_men\_aux"></param>

/// <param name="email\_men\_aux"></param>

/// <param name="altemail\_men\_aux"></param>

/// <returns></returns>

public bool Consist\_Fields(string name\_men\_aux, string last\_men\_aux,

string address\_men\_aux, string city\_men\_aux,

string province\_men\_aux, string postal\_men\_aux,

string age\_men\_aux, string pass1\_men\_aux,

string pass2\_men\_aux, string email\_men\_aux,

string altemail\_men\_aux)

{

bool Validation = true;

if (TextFirstName.Text == "")

{

Validation = false;

MyLabel\_First.Visible = true;

MyLabel\_First.Text = "Please insert your First Name";

}

//Check last Name filled by user

if (TextLastName.Text == "")

{

Validation = false;

MyLabel\_Last.Visible = true;

MyLabel\_Last.Text = "Please insert your Last Name";

}

//Check Address was filled by user

if (TextAddress.Text == "")

{

Validation = false;

MyLabel\_Address.Visible = true;

MyLabel\_Address.Text = "Please insert your Address";

}

//Check City filled by user

if (TextCity.Text == "")

{

Validation = false;

MyLabel\_City.Visible = true;

MyLabel\_City.Text = "Please insert the City";

}

//Check Province filled by user

if (TextProvince.Text == "")

{

Validation = false;

MyLabel\_Province.Visible = true;

MyLabel\_Province.Text = "Please insert the Province";

}

else

{

bool Check\_Province = Check\_Province\_Canada(TextProvince.Text);

if (Check\_Province == false)

{

MyLabel\_Province.Visible = true;

MyLabel\_Province.Text = "Only Canadian Province with two characters";

}

}

//Check Postal Code was filled by user

if (TextPostalCode.Text == "")

{

Validation = false;

MyLabel\_Postal.Visible = true;

MyLabel\_Postal.Text = "Please insert the Postal Code";

}

else

{ //check post code - function in C# arq type aspx.cs

bool Check\_Potal\_ok = Check\_PostalCode(TextPostalCode.Text);

if (Check\_Potal\_ok == false)

{

Validation = false;

MyLabel\_Postal.Text = "It is not a valid Canadian Postal Code - xxx-xxx";

MyLabel\_Postal.Visible = true;

}

}

//Check age filled by user

if (TextAge.Text == "")

{

Validation = false;

MyLabel\_Age.Visible = true;

MyLabel\_Age.Text = "Please insert your Age";

}

else

{

//Check if age is valid

int Ws\_Age\_number = Convert.ToInt32(TextAge.Text);

bool Check\_Age\_ok = Validade\_Age(Ws\_Age\_number);

if (Check\_Age\_ok == false)

{

Validation = false;

}

}

//Check password was filled by user

if (TextPassword.Text == "")

{

Validation = false;

MyLabel\_Password.Visible = true;

MyLabel\_Password.Text = "Please insert a password";

}

//Check password conrfirm was filled by user

if (TextPasswordConfirmation.Text == "")

{

Validation = false;

MyLabel\_PasswordConfirmation.Visible = true;

MyLabel\_PasswordConfirmation.Text = "Please insert the same password";

}

//check if it is a valid password - function in C# arq type aspx.cs

if (TextPassword.Text == TextPasswordConfirmation.Text)

{

bool Check\_Pass\_ok = Check\_Password(TextPassword.Text, TextPasswordConfirmation.Text);

if (Check\_Pass\_ok == false)

{

Validation = false;

}

}

//Check email filled by user

if (TextEmail.Text == "")

{

Validation = false;

MyLabel\_Email.Visible = true;

MyLabel\_Email.Text = "Please insert your e-mail";

}

else

{

bool Check\_Email\_ok = Check\_Email(TextEmail.Text);

if (Check\_Email\_ok == false)

{

Validation = false;

MyLabel\_Email.Visible = true;

MyLabel\_Email.Text = "Please insert a valid email address";

}

}

if (TextEmail\_Alternative.Text != "")

{

bool Check\_Email\_Alt\_ok = Check\_Email(TextEmail\_Alternative.Text);

if (Check\_Email\_Alt\_ok == false)

{

Validation = false;

MyLabel\_EmailAlt.Visible = true;

MyLabel\_EmailAlt.Text = "Please insert a valid email address";

}

else

{

if (TextEmail.Text == TextEmail\_Alternative.Text)

{

Validation = false;

MyLabel\_EmailAlt.Visible = true;

MyLabel\_EmailAlt.Text = "Alternative Email must be different from Email";

}

}

}

return (Validation);

}

/// <summary>

/// Register button. Check if all fields are fill in and check if it is a new student

/// </summary>

/// <param name="s"></param>

/// <param name="e"></param>

public void ButtonProcessar\_Click(Object s, EventArgs e)

{

MyLabel\_First.Visible = false;

MyLabel\_Last.Visible = false;

MyLabel\_Age.Visible = false;

MyLabel\_Address.Visible = false;

MyLabel\_City.Visible = false;

MyLabel\_Password.Visible = false;

MyLabel\_EmailAlt.Visible = false;

MyLabel\_Postal.Visible = false;

MyLabel\_Province.Visible = false;

MyLabel\_Email.Visible = false;

MyLabel\_MenssageResult.Visible = false;

MyLabel\_PasswordConfirmation.Visible = false;

bool Validation\_ok = Consist\_Fields(TextFirstName.Text, TextLastName.Text,

TextAddress.Text, TextCity.Text,

TextProvince.Text, TextPostalCode.Text,

TextAge.Text, TextPassword.Text,

TextPasswordConfirmation.Text,

TextEmail.Text,

TextEmail\_Alternative.Text);

//Verify if all register are right to show success message

if (Validation\_ok == true)

{

//Show the success message

MyLabel\_MenssageResult.Visible = true;

string Student\_Full\_Name = string.Concat(TextFirstName.Text," ", TextLastName.Text);

MyLabel\_MenssageResult.Text = "Thanks " + Student\_Full\_Name + " for registering with our website, your record was created successfully";

while (MyCheck\_AnotherMember.Checked == true && Validation\_ok == true)

{

Clean\_TextBoxes();

lblMensagem\_Quant\_Regs.Text = "Please register a new student";

MyLabel\_MenssageResult.Visible = false;

MyCheck\_AnotherMember.Checked = false;

string Last\_Reg\_FirstName = TextFirstName.Text;

string Last\_Reg\_LastName = TextLastName.Text;

string Last\_Reg\_TextAddress = TextAddress.Text;

string Last\_Reg\_TextCity = TextCity.Text;

string Last\_Reg\_TextProvince = TextProvince.Text ;

string Last\_Reg\_TextPostalCode = TextPostalCode.Text ;

string Last\_Reg\_TextAge = TextAge.Text;

string Last\_Reg\_TextPassword = TextPassword.Text ;

string Last\_Reg\_TextPasswordConfirmation = TextPasswordConfirmation.Text;

string Last\_Reg\_TextEmail = TextEmail.Text;

string Last\_Reg\_TextEmailAlternative = TextEmail\_Alternative.Text;

string Last\_Reg\_TextPhone = TextPhone.Text;

}

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text.RegularExpressions;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace assignment01

{

public partial class form\_assignment : System.Web.UI.Page

{

///<summary>

///The method Check\_Password, cheks the password format and

///check the first password is equal the confirmation's password

///<param name="Check\_Password\_Aux"></param>

///<param name="Check\_Password\_Aux\_Alt"></param>

///</summary>

public bool Check\_Password(string Check\_Password\_Aux, string Check\_Password\_Aux\_Alt)

{

//Regular expression to valid email

//At least one upper case english letter, (?=.\*?[A - Z])

//At least one lower case english letter, (?=.\*?[a - z])

//At least one digit, (?=.\*?[0 - 9])

//At least one special character, (?=.\*?[#?!@$%^&\*-])--- I do not use it in this example

//Minimum 6 in length.{6,} (with the anchors

string Pattern = "^(?=.\*?[A-Z])(?=.\*?[a-z])(?=.\*?[0-9]).{6,}$";

Regex reg = new Regex(Pattern, RegexOptions.IgnoreCase | RegexOptions.Compiled);

if (!(reg.IsMatch(Check\_Password\_Aux) && reg.IsMatch(Check\_Password\_Aux\_Alt)))

{

MyLabel\_Password.Text = "Passwords must have at least 6 characters and must contain at"+

"least one digit and one upper-case character";

MyLabel\_Password.Visible = true;

MyLabel\_PasswordConfirmation.Text = "Passwords must have at least 6 characters and must contain at" +

"least one digit and one upper-case character";

MyLabel\_PasswordConfirmation.Visible = true;

return false;

}

else

if (Check\_Password\_Aux != Check\_Password\_Aux\_Alt)

{

MyLabel\_Password.Text = "ATENTION - Password doens't match";

MyLabel\_Password.Visible = true;

MyLabel\_PasswordConfirmation.Text = "ATENTION - Password doens't match";

MyLabel\_PasswordConfirmation.Visible = true;

return false;

}

else

return true;

}

///<summary>

///The method Check\_PostalCode, cheks the postal code format and

///if it is Canadian

///<param name="TextPostalCode\_aux"></param>

///</summary>

public bool Check\_PostalCode(string TextPostalCode\_aux)

{

//Regular expression to valid Canadian Postal Code in the format of "M3A-1A5"

//Character Capital Letter: [ABCEGHJ-NPRSTVXY]

//Quantity{1}

//Number range [0-9]

//[-] separetor

string Pattern = "^[ABCEGHJ-NPRSTVXY]{1}[0-9]{1}[ABCEGHJ-NPRSTV-Z]{1}[-]?[0-9]{1}[ABCEGHJ-NPRSTV-Z]{1}[0-9]{1}$";

Regex reg = new Regex(Pattern, RegexOptions.IgnoreCase | RegexOptions.Compiled);

if (!(reg.IsMatch(TextPostalCode\_aux)))

{

return false;

}

return true;

}

protected void TextProvince\_TextChanged(object sender, EventArgs e)

{

TextProvince.Text = TextProvince.Text.ToUpper();

}

}