

Lab 16 – Security Fundamentals

Objective

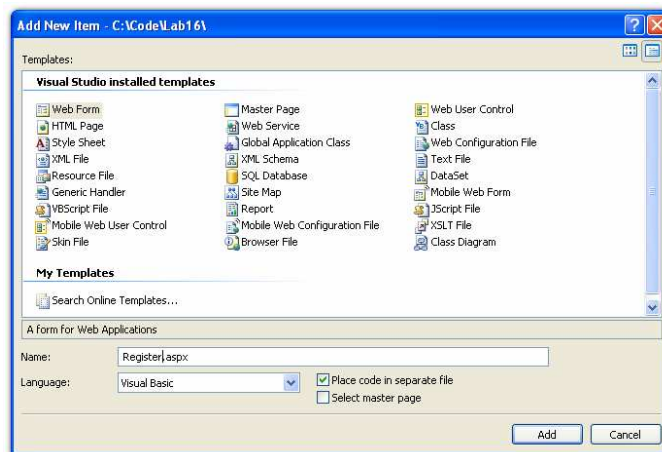
You will create an application that will be restricted to only logged in users. This application will have a registration page that will be used to create a single user account in a local XML file. A login page will also be created to verify a typed in user name and password prior to letting the user into the site.

Step-by-Step Instructions

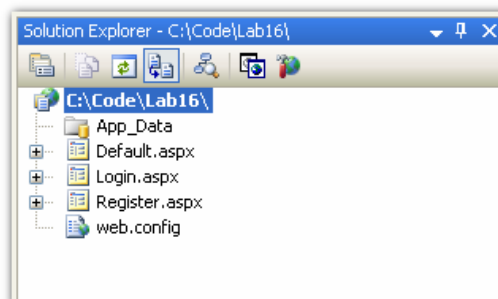
1. Start Visual Studio 2005
2. Create a new web site. **File** → **New** → **Web Site** (Or, **File** → **New Web Site...**)
3. Select the following options:

Setting	Value
Visual Studio installed templates	ASP.NET Web Site
Location	File System – C:\Code\Lab16
Language	Visual Basic

4. From the **WebSite** menu choose **Add New Item...**
5. Create a new Web Form named *Register.aspx* as shown below:



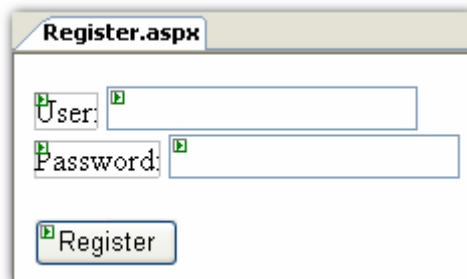
6. Repeat steps 4 and 5 above to create a form named *Login.aspx*. The project should now look similar to the one below:



7. Open *Register.aspx*.

8. Add controls to the *Register.aspx* as shown below. You can add these controls in Design view or Source view – which ever you are more comfortable with. Make sure, however, that the control names and properties are set according to the code sample shown in the Source view shown below.

Note that the *TextMode* property for the *txtPassword* control is set to “password”. This will mask the password typed in by the user.



```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Register.aspx.vb" Inherits="Register" %>
<!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">
    <asp:Label ID="Label1" runat="server" Text="User:"></asp:Label>
    <asp:TextBox ID="txtUser" runat="server"></asp:TextBox>
    <br />
    <asp:Label ID="Label2" runat="server" Text="Password:"></asp:Label>
    <asp:TextBox ID="txtPassword" runat="server"
      TextMode="Password"></asp:TextBox>
    <br />
    <br />
    <asp:Button ID="cmdRegister" runat="server" Text="Register" />
  </form>
</body>
</html>
```

9. Save file *Register.aspx*.
10. Create the click event of the *cmdRegister* button.

11. Place the following code in this click event. This code will create an XML file that will store your user database. There will be a single entry placed in the XML file for the user name and password that is typed into the form.

```
Imports System.Xml

Partial Class Register
    Inherits System.Web.UI.Page

    Protected Sub cmdRegister_Click(ByVal sender As Object, ByVal e As System.EventArgs) _
        Handles cmdRegister.Click

        Dim doc As New XmlDocument()
        Dim RootElement As XmlElement
        Dim UserElement As XmlElement
        Dim UserAttribute As XmlAttribute

        Dim Declaration As XmlDeclaration
        Declaration = doc.CreateXmlDeclaration("1.0", Nothing, "yes")

        doc.InsertBefore(Declaration, doc.DocumentElement)

        RootElement = doc.CreateElement("UserList")
        doc.InsertAfter(RootElement, Declaration)

        UserElement = doc.CreateElement("User")
        RootElement.AppendChild(UserElement)

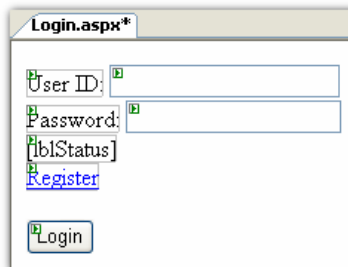
        UserAttribute = doc.CreateAttribute("UserID")
        UserAttribute.Value = txtUser.Text
        UserElement.SetAttributeNode(UserAttribute)

        UserAttribute = doc.CreateAttribute("Password")
        UserAttribute.Value = txtPassword.Text
        UserElement.SetAttributeNode(UserAttribute)

        doc.Save("c:\UserList.xml")

        Response.Redirect("~/login.aspx")
    End Sub
End Class
```

12. Save and close *Register.aspx*.
13. Open *Login.aspx*.
14. Add controls to the *Login.aspx* as shown below. You can add these controls in Design view or Source view – which ever you are more comfortable with. Make sure, however, that the control names and properties are set according to the code sample shown in the Source view shown below.



The screenshot shows a web form titled "Login.aspx*" with the following controls:

- A text input field labeled "User ID:".
- A text input field labeled "Password:".
- A label control with the text "[lblStatus]".
- A blue hyperlink labeled "Register".
- A button labeled "Login".

```

<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Login.aspx.vb" Inherits="Login" %>

<!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">
    <asp:Label ID="Label1" runat="server" Text="User ID:"></asp:Label>
    <asp:TextBox ID="txtUser" runat="server"></asp:TextBox>
    <br />
    <asp:Label ID="Label2" runat="server" Text="Password:"></asp:Label>
    <asp:TextBox ID="txtPassword" runat="server" TextMode="Password"></asp:TextBox>
    <br />
    <asp:Label ID="lblStatus" runat="server"></asp:Label>
    <br />
    <asp:LinkButton ID="lnkRegister" runat="server"
      PostBackUrl="~/register.aspx">Register</asp:LinkButton>
    <br />
    <br />
    <asp:Button ID="cmdLogin" runat="server" Text="Login" />
  </form>
</body>
</html>

```

15. Create the click event on the *cmdLogin* button.
16. Add the following code to the click event. This code will open the *UserList.xml* file created in the *Register.aspx* and verify the user and password against the values entered by the user into the *Login.aspx* screen.

```

Imports System.Xml

Partial Class Login
  Inherits System.Web.UI.Page

  Protected Sub cmdLogin_Click(ByVal sender As Object, ByVal e As System.EventArgs) _
    Handles cmdLogin.Click

    Dim doc As New XmlDocument()
    doc.Load("c:\UserList.xml")

    Dim Element As XmlElement

    For Each Element In doc.DocumentElement.ChildNodes
      If Element.GetAttribute("UserID") = txtUser.Text And _
        Element.GetAttribute("Password") = txtPassword.Text Then

        FormsAuthentication.RedirectFromLoginPage(txtUser.Text, False)
      Else
        lblStatus.Text = "Invalid Login"
      End If
    Next
  End Sub
End Class

```

17. Save and close *Login.aspx*
18. Open *Default.aspx*.
19. Place a **Label** control on the form.
20. Create the *Page_Load* event.

21. Add the following code to the *Page_Load* event. This code will place the user login name in the text once the user is logged in to the web site.

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
    Label1.Text = "Welcome: " & User.Identity.Name
End Sub
```

22. Save and close *Default.aspx*.
23. Open the *web.config* file.
24. Edit the *web.config* as shown below. These changes should be placed at the end of the file before the closing `</configuration>` element.

The authentication node setting is changed to *Forms* from *Windows*. This change is necessary because we do not want to use authentication based upon the Windows accounts on the web server. We want the authentication to come from the XML file created in the lab.

Along with the Forms authentication, we will add the *loginUrl* attribute to the *forms* element to point to the *login.aspx* page. This pointer is important and will be used when users request a page before they are authenticated.

The `<authorization>` element will need to be added to deny all anonymous users from accessing the site. This element will make the web site require a successful login.

The `<location path="register.aspx">` section is added to make sure that anonymous users can get to the *register.aspx* page. This way, the user can navigate to the *register.aspx* to register on the web site. The `<allow users="*" />` is used here to signify that any anonymous user can get to this page

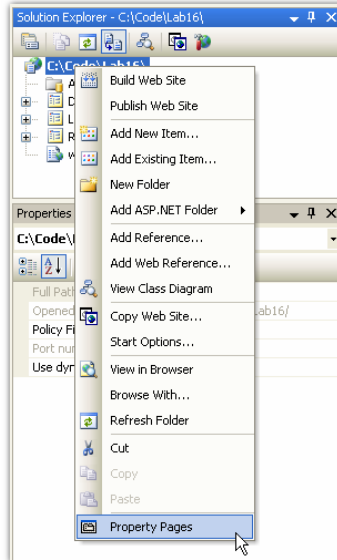
Note: the text below is case sensitive.

```
<authentication mode="Forms">
  <forms loginUrl="login.aspx"></forms>
</authentication>
<authorization>
  <deny users="?" />
</authorization>
</system.web>

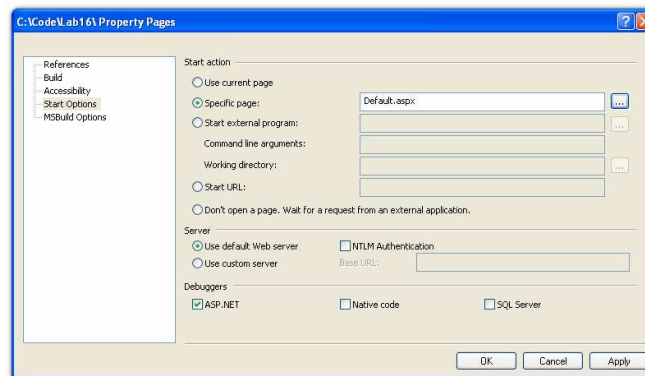
<location path="register.aspx">
  <system.web>
    <authorization>
      <allow users="*" />
    </authorization>
  </system.web>
</location>
</configuration>
```

25. Save and close the *web.config* file.

26. Set the start page on the web site by right-clicking on the project and choosing **Property Pages**:



27. Select **Start Options**, then **Specific page:**, then type “*Default.aspx*” in the box.



28. Click **OK** to set this option.
29. Run the application.
30. You will be brought to the **Login** page.
31. Since you don't have an account, click on the **Register** link.
32. Enter your first name and a password into the form and then click the **Register** button. Your first name and password will now be stored in the *Userlist.xml* file. You will then be redirected back to the **Login** page.
33. Enter your first name and password into the *login.aspx* form and click the **Login** button.
34. You will now be taken to *default.aspx* since you have successfully logged in.

