

### Assignment #3

**Objective:** Learn about HTML forms, server-side processing (using Microsoft’s ASP.NET Web Pages), implement user registration and login. Other techniques learned include DB store and lookup, and using permanent cookies.

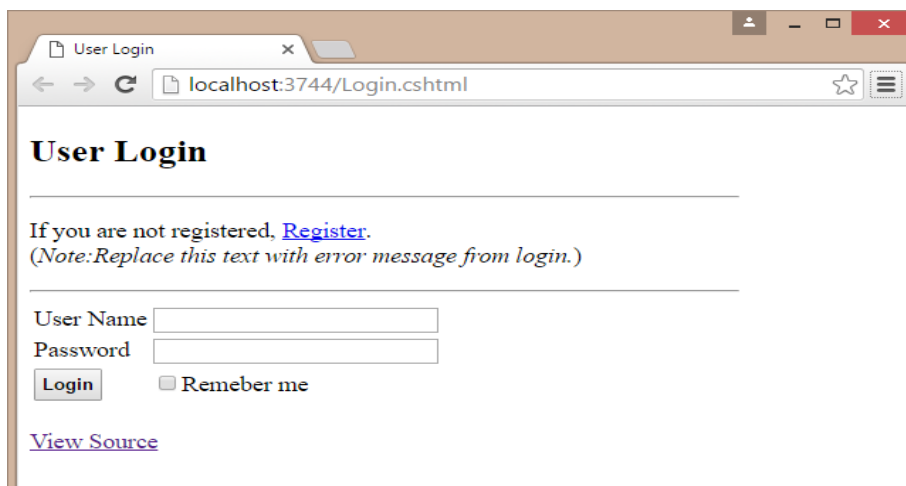
**Submission & Deadline:** Oct. 16, 2016 - send an e-mail to instructor with the URL of your site.

You are to develop a basic web application. The application’s “Default.cshtml” page automatically displays “Welcome <UserName>” if the incoming HTTP request includes a cookie named “UserID” with a valid user id (use the UserID to lookup UserName from DB); otherwise, the user is redirected (use *Response.Redirect*) to the login page. The application includes two more web pages: “Login.cshtml” and “Register.cshtml”. *The login page has a link to the registration page.* The login page is shown below where you have to supply the server-side code to process the submitted data as follows:

If login information is invalid (based on DB lookup) then reply with the login page again with a proper message at top of the page (either *invalid user\_name* or *invalid password*); otherwise, redirect to the default page; in addition, if the user has checked “Remember me” then send a permanent cookie (*UserID=<UserID>*).

The user’s registration information is stored into MS SQL CE database (use a DB table named “tblUsers”) when registration form is submitted. It consists of three fields: *UserID*, *UserName*, *Passwd*, where UserID (set as Primary Key) is a string of 8 random digits which is generated by the function `genUserID()` (see below) .

**Note:** You can test the login page by entering dummy data into your DB directly.



The screenshot shows a web browser window with the title "User Login" and the address bar showing "localhost:3744/Login.cshtml". The page content is as follows:

**User Login**

---

If you are not registered, [Register](#).  
(Note: Replace this text with error message from login.)

---

User Name   
Password

Remember me

[View Source](#)

### Recommended Readings on Razor

1. <http://www.asp.net/web-pages/tutorials/basics/2-introduction-to-asp-net-web-programming-using-the-razor-syntax>
2. <http://www.asp.net/web-pages/overview/data/5-working-with-data>
3. Do Google search for “*tutorial razor filetype: pdf*”  
The first hit (*download.microsoft.com*) is a link to a good reference on WebMatix and Razor.

## Login.cshtml

```
<!DOCTYPE html>
<html>
<head><title>User Login</title>
<style>
  #myFrame { display:none; width:550px; height:350px; overflow:auto; border:1px solid gray }
</style>

</head>
<body style="width:500px;font-size: 1.1em">
<h2>User Login</h2>
<form method="GET" action="Login.cshtml" >
<input type="hidden" name="hvar" value="hval" >
<hr>
<p>If you are not registered, <a href="Register.cshtml" >Register</a>.
  <br><em>Note:Replace this text with error message from login.</em>
</p>
<hr>
<table>
<tr><td>User Name</td><td><input type="text" name="UserName" size="24" > </td></tr>
<tr><td>Password</td><td><input type="text" name="Passwd" size="24" > </td></tr>
<tr><td><input type="Submit" value="Login" style="padding:4px;font-weight:bold" ></td>
<td><input type="CheckBox" name="ChkRemember" >Remeber me</td></tr>
</table>
</form>
<br>
<a href="ViewSrc.cshtml" target="myFrame"
  onclick="document.getElementById('myFrame').style.display='block'" >View Source</a>
<iframe id="myFrame" name="myFrame" ></iframe>
</body>
</html>
```

The “ViewSrc.cshtml” script reads the file “Login.cshtml” and sends all of its content to the client. It uses the following code:

```
@{
    Response.ContentType = "text/plain";
    // OR USE Response.AddHeader("Content-Type", "text/plain");

    var FilePath = Server.MapPath("Login.cshtml");
    var txt = File.ReadAllText(FilePath);

    Response.Write(txt);
    Response.End();
}
```

**Note:** The *content-type* in the preceding script is set to “text/plain” (instead of “text/html”) so that the browser does not interpret the text as HTML.

```
@functions
{
    // --- Use the following function to generate a User ID
    string genUserID()
    {
        Random rand = new Random();
        string str = "";
        for (int i = 1; i <= 8; i++)
        {
            str = str + rand.Next(10);
        }
        return str;
    }
}
```