

Overview: Use ASP.NET forms to gather user input & generate dynamic output

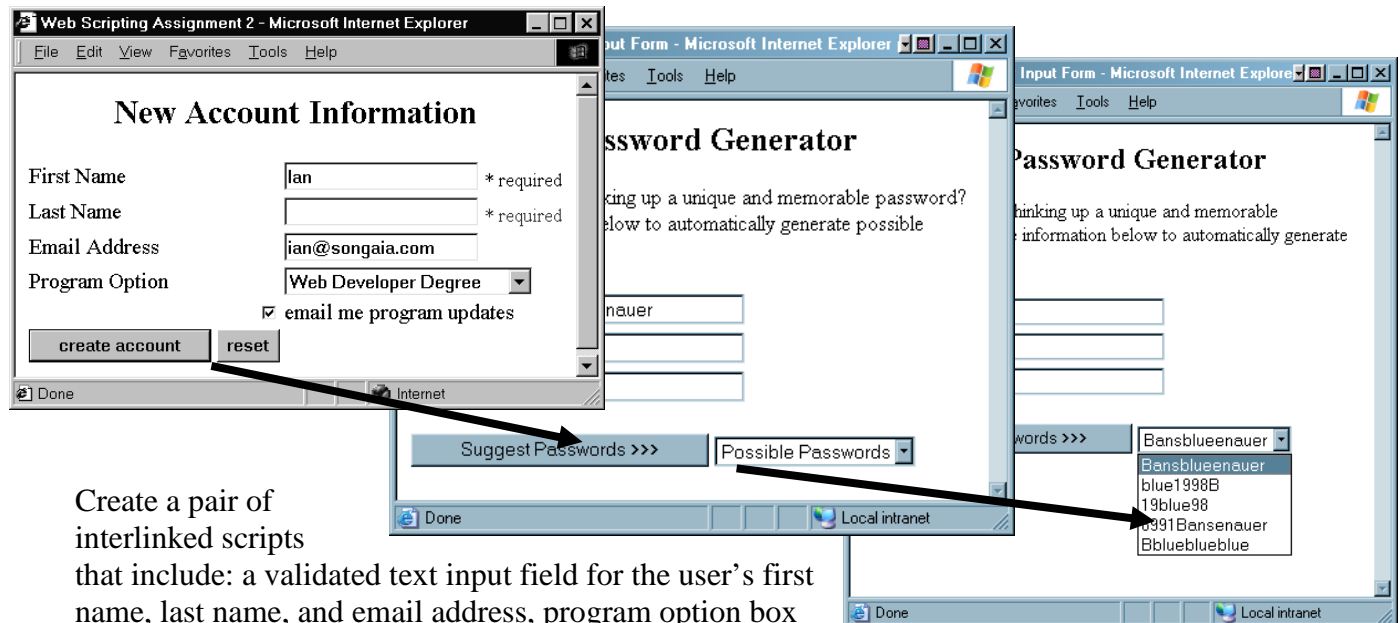
Part 1: ASP.NET Form using String functions, arrays and data binding

Create an ASP.NET Form with text input fields for the user's last name, birth year and favorite color. When the password form is submitted, by clicking on the **Suggest Passwords** button, invoke an ASP script that creates an HTML page showing five suggested passwords. Use String functions to display five unique password suggestions.

Extra credit:

The image is a side-by-side comparison of a web form titled 'Automatic Password Generator'. On the left, the form has three input fields: 'Last Name:' with 'Bansenauer', 'Birth Year:' with '1998', and 'Favorite Color:' with 'blue'. Below these is a button labeled 'Suggest Passwords >>>' and a dropdown menu labeled 'Possible Passwords'. On the right, the same form is shown after the button was clicked. The dropdown menu is now open, displaying a list of generated passwords: 'Bansblueenauer', 'blue1998B', '19blue98', '8991Bansenauer', and 'Bblueblueblue'. A large black arrow points from the button on the left to the dropdown menu on the right, with the text 'Button Click' written above it.

Part 3: Build a set of interlinked scripts (using validated web controls)



Create a pair of interlinked scripts that include: a validated text input field for the user's first name, last name, and email address, program option box (include Web Developer Degree, ETSP Degree, Network Technology Degree) and a confirmation email checkbox.

The New Account Information form verifies the data before submitting the form according to the following: the first and last name fields are required and if the checkbox is selected, the email address is also required. The Password Generator form validates that no item is left blank and that the birth year is a four digit number.

Assign each item of data collected to a Session variable. Use the session information to provide a more personal interaction with the user (e.g., include their first name in text of your password generator, etc.)

Part 4: A Visitor Log Application

Modify your *login.asp* ASP file so that when a user successfully logs in the script generates a Visitor Log table as shown below. *At this point, successfully logging in amounts to nothing more than filling in any username and password and pressing "Login."*

The Visitor Log table should include the current date and time (client side) at the top of the page, a welcome message with the users login name and a table showing all current users; the user name, login date and time (server side) and IP address – this last item is available in the Request object's ServerVariables collection.

Visitor Log: Use Case

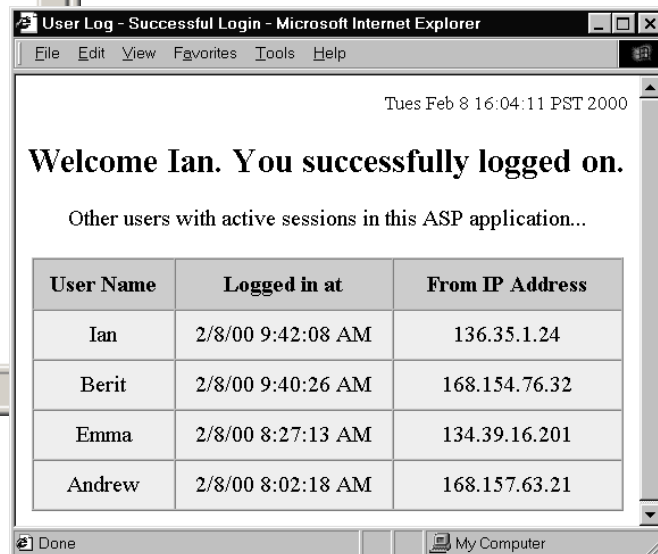
The user logs into the application with **ANY** username and password combination and views a table that displays all the other visitors that are currently logged into the application. The user refreshes the display periodically to check who else may have recently logged in. The user logs out when finished.



global.asax file (see attached) initializes the *visitorTable* to hold the session id, user name, login time, and IP address in the following format.

<i>Column 0</i>	<i>Column 1</i>	<i>Column 2</i>	<i>Column 3</i>
session id	user name	login time	IP address

1. Your script should use either an **Application variable** called *visitorTable* stored as a Data Table or an XML document called *visitor.xml* stored in **App_Data**. A



2. Provide a refresh button that updates the display of the current data
3. Provide a logout button that removes the user's entry from the Application table and displays a message confirming logout. The user should then be prompted to log-in again.
4. **(BONUS)** Provide a button that writes the *visitorTable* data to a log file in XML format. The log file would include the date and time in its filename.
5. **(BONUS)** Others ideas that fit conceptually in this Use Case.

Part of the global.asax file for the Application

```
<%@ Application Language="C#" %>
<%@ Import Namespace="System.Data"%>

<script runat="server">
/**
 * Initialize a new DataTable variable for the application,
 * define the data types in its columns
 */
void Application_Start(object sender, EventArgs e)
{
    DataTable dt = new DataTable();
    dt.Columns.Add(new DataColumn("session_id",
System.Type.GetType("System.String")));
    dt.Columns.Add(new DataColumn("username",
System.Type.GetType("System.String")));
    dt.Columns.Add(new DataColumn("login_time",
System.Type.GetType("System.DateTime")));
    dt.Columns.Add(new DataColumn("ip_address",
System.Type.GetType("System.String")));

    // store the DataTable as an Application variable
    Application["visitorTable"] = dt;
}

</script>
```

XML document format

```
<?xml version="1.0" standalone="yes"?>
<VisitorsLog>
  <visitor>
    <sessionID>960002</sessionID>
    <userName>Ian</userName>
    <loginTime>1/2/10 9:42:08 AM</loginTime>
    <ipAddress>132.35.1.24</ipAddress>
  </visitor>
</VisitorsLog>
```

Assignment 1 Submission

Name:

Email:

Assess yourself using the evaluation scale, suggestion and comment space below. TOTAL SCORE: /16

Evaluation: 0 – Absent 1 – Effort 2 – Developing 3 – Adequate 4 – Well Developed

Write Supporting Code for Web Applications

comments

- Code meets is designed so app performs efficiently
 - All ASP scripts use appropriate methods and redirection
 - Scripts appropriately and efficiently use variables.
- Code meets project **Login** Use Case functional specs
 - Submitting the login form with correct credentials causes the welcome page to be displayed
 - Entering any other username and password combination displays a login error message.
 - Selecting create new account or submitting the login form without a password displays the password generator page.
 - Submitting the password generator form creates five unique password suggestions from concatenations of user input.
- Code meets project **Visitor Log** Use Case functional specs
 - Submitting the login form with correct credentials causes the welcome page to be displayed
 - Entering any other username and password combination displays a login error message.
 - Selecting create new account or submitting the login form without a password displays the password generator page.
 - Submitting the password generator form creates five unique password suggestions from concatenations of user input.
- Code meets project objectives and functional specs
 - Data validation works as specified in Part 1
 - Visitor Log Application displays all current users.
 - Visitor Log Application correctly adds new users.
 - Visitor Log Application appropriately logs out users.
- Code properly documented to ensure maintainability
 - Comment lines include name, the assignment number, the date, and a brief description of the script's purpose
 - Clearly written comments document the purpose of each function and any other code segment as needed.
 - Scripts use meaningful variable and method names and clear, consistent indentation