<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="Testing.WebForm2" %>

<!DOCTYPE html>

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

<head id="Head1" runat="server">

<title>Birthday Paradox App</title>

<style>

body {background-color: linen;}

h1, h2 {color: maroon;}

.btn { background-color: #C9E2E9; border:ridge; padding: 15px 32px; text-align: center; text-decoration: none; display: inline-block;

font-size: 16px; margin: 4px 2px; cursor: pointer; -webkit-transition-duration: 0.4s; transition-duration: 0.4s;}

.btn1:hover {box-shadow: 0 12px 16px 0 rgba(0,0,0,0.24),0 17px 50px 0 rgba(0,0,0,0.19);}

</style>

</head>

<body>

<form id="form1" runat="server">

<h2>Click the button to begin process</h2>

<div><p><asp:Button CssClass="btn btn1" ID="startProcess" runat="server" OnClick="clickedButton" Text="Start Process" BorderStyle="NotSet" /></p></div>

<div><asp:GridView ID="GridView1" runat="server" CssClass="Grid" AutoGenerateColumns="false">

<Columns>

<asp:BoundField DataField="Count" HeaderText="Count" ItemStyle-Width="120" />

<asp:BoundField DataField="Matched Date" HeaderText="Matched Date" ItemStyle-Width="120" />

<asp:BoundField DataField="Attempts" HeaderText="Attempts" ItemStyle-Width="120" />

</Columns>

</asp:GridView></div>

</form>

</body>

</html>

using System;

using System.Data;

using System.Collections;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Testing

{

public partial class WebForm2 : System.Web.UI.Page

{

Random gen = new Random();

string \_dateMatched;

int \_numberGenerated;

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

{

GenerateResults();

if (!this.IsPostBack)

{

DataTable tblResults = new DataTable();

tblResults.Columns.AddRange(new DataColumn[3] { new DataColumn("Count"), new DataColumn("Matched Date"), new DataColumn("Attempts") });

ViewState["tableResults"] = tblResults;

this.BindGrid();

}

}

protected void BindGrid()

{

GridView1.DataSource = (DataTable)ViewState["tableResults"];

GridView1.DataBind();

}

private void GenerateResults()

{

DateTime dtGenDate2;

int nLoopCounter = 0;

string sGenDate1 = GetRandomDate().ToShortDateString();

string sGenDate2 = null;

IList dateList = new ArrayList();

dateList.Add(sGenDate1);

do

{

dtGenDate2 = GetRandomDate();

sGenDate2 = dtGenDate2.ToShortDateString();

nLoopCounter++;

if (dateList.Contains(sGenDate2))

{

break;

}

else

{

dateList.Add(sGenDate2);

}

} while (nLoopCounter < 1000);

\_dateMatched = sGenDate2;

\_numberGenerated = nLoopCounter;

}

public DateTime GetRandomDate()

{

return DateTime.Now.AddDays(gen.Next(365));

}

protected void clickedButton(object sender, EventArgs e)

{

int nProcessCount;

try

{

nProcessCount = (int)ViewState["Counter"];

}

catch (Exception)

{

nProcessCount = 0;

}

nProcessCount++;

ViewState["Counter"] = nProcessCount;

DataTable tblResults = (DataTable)ViewState["tableResults"];

tblResults.Rows.Add(\_numberGenerated.ToString(), \_dateMatched.Trim(), nProcessCount);

ViewState["tableResults"] = tblResults;

this.BindGrid();

}

}

}