<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="RandomDate.aspx.cs" Inherits="WebApp.RandomDate" %>

<!DOCTYPE html>

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

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

<title>Birthday Paradox App</title>

<style>

html {

display: block;

}

.bold {

font-weight:bold;

}

table {

width: 100%;

table-layout: auto;

empty-cells: show;

}

.btn {

border: 0 solid transparent;

background-color: #fe8652;

border-radius: 3px;

padding: 0;

min-width: 160px;

text-transform: uppercase;

color: #fff;

font-weight: bold;

font-size: 14px;

font-family: "Open Sans";

height: 34px;

padding: 0 15px;

}

.Grid th {

text-align: left;

font-size: 14px;

font-family: "Open Sans";

height: 30px;

background: whitesmoke;

text-decoration: none;

border: 1px solid #cdcdcd;

padding-left: 10px;

padding-top: 5px;

padding-bottom: 4px;

font-weight: normal;

cursor: default;

}

.Grid td {

line-height: 23px;

padding-left: 5px;

padding: 0 5px;

height: 0;

}

</style>

</head>

<body>

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

<div><p><asp:Button CssClass="btn" 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" />

<asp:BoundField DataField="Matched Date" HeaderText="Matched Date" />

<asp:BoundField DataField="Attempts" HeaderText="Attempts" />

</Columns>

</asp:GridView>

</div>

</form>

</body>

</html>

using System;

using System.Collections;

using System.Data;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace WebApp

{

public partial class RandomDate : 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("The app tried " + \_numberGenerated.ToString() + " times", "Found this matched date: " + \_dateMatched.Trim(), "Button clicked " + nProcessCount + " times");

ViewState["tableResults"] = tblResults;

this.BindGrid();

}

}

}