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

<!DOCTYPE html>

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

<head runat="server">

<title>Birthday Paradox</title>

<style>

html {

display: block;

}

h2 {

text-indent: 1%;

color: #354140;

font-size: 15px;

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;

}

.footer {

position: fixed;

left: 0;

bottom: 0;

width: 100%;

background-color: whitesmoke;

padding: 20px;

}

</style>

</head>

<body>

<div>

<h2>When you press the start procces button, the application will generate random dates until a match is found.

Then it will output the data to a table each time you press the button.</h2>

</div>

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

<div>

<p>

<asp:Button CssClass="btn" ID="startProcess" runat="server" OnClick="clickedButton" Text="Start Process" />

</p>

</div>

<div>

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

<Columns>

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

<asp:BoundField DataField="Num of Dates Generated" HeaderText="Num of Dates Generated" />

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

</Columns>

</asp:GridView>

</div>

<div class="footer">

Average for Num of Dates Generated:

<asp:Label ID="lblavgNum" runat="server" Font-Size="Larger" Text=""></asp:Label>

</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 WebApp

{

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

{

Random gen = new Random();

string \_dateMatched;

int \_numberGenerated;

private double Average(DataTable tblResults)

{

double dAvg = 0;

double dSum = 0;

double rowCount = tblResults.Rows.Count;

foreach (DataRow row in tblResults.Rows)

{

dSum += Convert.ToInt32(row[1]);

dAvg = dSum / rowCount;

dAvg = Math.Round(dAvg, 2);

}

return dAvg;

}

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

{

GenerateResults();

if (!this.IsPostBack)

{

DataTable tblResults = new DataTable();

tblResults.Columns.AddRange(new DataColumn[3] { new DataColumn("Run"), new DataColumn("Num of Dates Generated"), new DataColumn("Matched Date") });

ViewState["tableResults"] = tblResults;

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(nProcessCount, \_numberGenerated.ToString(), \_dateMatched.Trim());

ViewState["tableResults"] = tblResults;

BindGrid();

double dAvgNum = Average(tblResults);

lblavgNum.Text = dAvgNum.ToString();

}

}

}