//CalcController response to "C is for Controller"

using System;

using System.Collections.Generic;

using System.Diagnostics;

using System.Linq;

using System.Threading.Tasks;

using Microsoft.AspNetCore.Mvc;

using MVCManukauTech.Models;

//20180226 JPC JSON component - if this gives errors, it may need downloading with "NuGet"

using Newtonsoft.Json;

namespace MVCManukauTech.Controllers

{

public class CalcController : Controller

{

// GET: Calc

public ActionResult Index()

{

return View();

}

//Test URL

//GET /calc/add?a=4&b=11

public string Add(double a, double b)

{

double c = a + b;

return c.ToString();

}

//Test URL

//GET /calc/quad?a=1&b=6&c=9

//20180226 JPC needs type IActionResult instead of string to return HTML

public IActionResult Quad(double a, double b, double c)

{

//Needs this statement with return Content - not needed with return View

Response.ContentType = "text/html";

//coding of the supplied formula goes here

double x1;

double x2;

x1 = (-b + Math.Sqrt(b \* b - 4 \* a \* c)) / (2 \* a);

x2 = (-b - Math.Sqrt(b \* b - 4 \* a \* c)) / (2 \* a);

string response = "The 2 solutions are <b>" + x1 + "</b> and <b>"

+ x2 + "</b>";

return Content(response);

}

//Test URL for an expected mean of 5

//GET /calc/mean?csv=4,4,4,4,5,6,6,6,6

public IActionResult Mean(string csv)

{

Response.ContentType = "text/html";

double total = 0;

double mean;

string[] values = csv.Split(',');

for(int i=0; i < values.Length; i++)

{

total += Convert.ToDouble(values[i]);

}

mean = total / values.Length;

string output = "<span style='font-size:30pt'>The mean is <span style='font-weight:bold;color:red'>"   
 + mean.ToString() + "</span></span>";

return Content(output);

}

//Test URL for an expected mean of 5

//GET /calc/jsonmean?json=[4,4,4,4,5,6,6,6,6]

public IActionResult JSONMean(string json)

{

Response.ContentType = "text/html";

//Deserialize to a double array - need to include that type in <T> format

double[] values = JsonConvert.DeserializeObject<double[]>(json);

double total = 0;

double mean;

for (int i = 0; i < values.Length; i++)

{

total += values[i];

}

mean = total / values.Length;

return Content(mean.ToString());

}

}

}