**Bahria University,**

**Karachi Campus**



**LAB EXPERIMENT NO.**

**\_8\_**

**LIST OF TASKS**

|  |  |
| --- | --- |
| TASK NO | **OBJECTIVE** |
| 01 | Develop a web service in ASP.NET to retrieve current weather data based on location. Then build an ASP.NET web application to consume the weather service and display current weather information for a given location. |
| 02 | Design and implement a web service that solves quadratic equations using different method (atleast 3), then build an ASP.NET web application to consume this service. |
|  |  |

**Submitted On:**

**Date: 25/April/2024**

**TASK # 1:** Develop a web service in ASP.NET to retrieve current weather data based on location. Then build an ASP.NET web application to consume the weather service and display current weather information for a given location.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Net;

using System.Security.Policy;

using System.Web;

using System.Web.Script.Serialization;

using System.Web.Services;

namespace lab\_8\_cc\_weather

{

[WebService(Namespace = "http://tempuri.org/")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

[System.ComponentModel.ToolboxItem(false)]

public class Weather : System.Web.Services.WebService

{

[WebMethod]

public string GetWeather(string location)

{

string apiKey = "380ce46778080a5b6283f1c6032bc714";

string apiUrl = $"https://api.openweathermap.org/data/2.5/weather?q={location}&appid={apiKey}";

try

{

using (WebClient client = new WebClient())

{

string jsonResponse = client.DownloadString(apiUrl);

JavaScriptSerializer serializer = new JavaScriptSerializer();

dynamic weatherData = serializer.Deserialize<dynamic>(jsonResponse);

string locationName = weatherData["name"];

double temperature = Convert.ToDouble(weatherData["main"]["temp"]);

string weatherDescription = weatherData["weather"][0]["description"];

temperature -= 273.15;

string weatherInfo = $"Weather in {locationName}: Temperature {temperature:F1}°C, {weatherDescription}";

return weatherInfo;

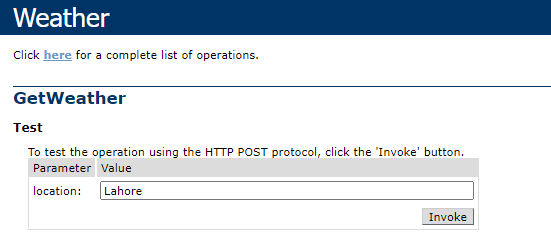
} }

catch (Exception ex)

{

return "Error fetching weather data: " + ex.Message;

} } } }



A screenshot of a computer

Description automatically generated

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Weather.aspx.cs" Inherits="lab\_8\_cc\_weatherdeploy.Weather" %>

<!DOCTYPE html>

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

<head runat="server">

<title>Weather Information</title>

</head>

<body>

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

<div>

<h1>Weather Information</h1>

<asp:Label ID="LabelLocation" runat="server" Text="Enter Location:"></asp:Label>

<asp:TextBox ID="TextBoxLocation" runat="server"></asp:TextBox>

<br />

<asp:Button ID="ButtonGetWeather" runat="server" Text="Get Weather" OnClick="ButtonGetWeather\_Click" />

<br />

<asp:Label ID="LabelWeatherInfo" runat="server" Text=""></asp:Label>

</div>

</form>

</body>

</html>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace lab\_8\_cc\_weatherdeploy

{

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

{

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

{

}

protected void ButtonGetWeather\_Click(object sender, EventArgs e)

{

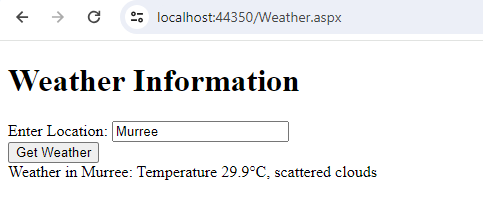
localhost.Weather service = new localhost.Weather();

string location = TextBoxLocation.Text.Trim();

string weatherInfo = service.GetWeather(location);

LabelWeatherInfo.Text = weatherInfo;

} } }



TASK # 2: Design and implement a web service that solves quadratic equations using different method (atleast 3), then build an ASP.NET web application to consume this service.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.Http;

using System.Web.Services;

namespace lab\_08\_task\_02

{

[WebService(Namespace = "http://tempuri.org/")]

[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1\_1)]

[System.ComponentModel.ToolboxItem(false)]

public class WebService1 : System.Web.Services.WebService

{

[WebMethod]

public double[] SolveQuadratic(double a, double b, double c)

{

double[] solutions = new double[2];

double discriminant = b \* b - 4 \* a \* c;

if (discriminant > 0)

{

solutions[0] = (-b + Math.Sqrt(discriminant)) / (2 \* a);

solutions[1] = (-b - Math.Sqrt(discriminant)) / (2 \* a);

}

else if (discriminant == 0)

{

solutions[0] = -b / (2 \* a);

solutions[1] = solutions[0];

}

return solutions;

}

[WebMethod]

public double[] SolveQuadraticByFactoring(double a, double b, double c)

{

throw new NotImplementedException();

}

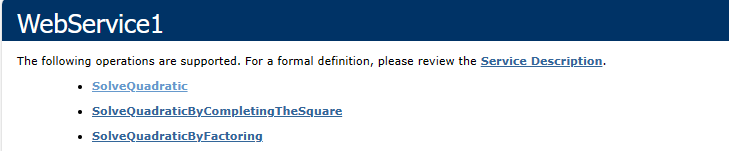
[WebMethod]

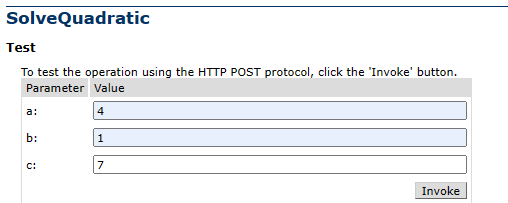
public double[] SolveQuadraticByCompletingTheSquare(double a, double b, double c)

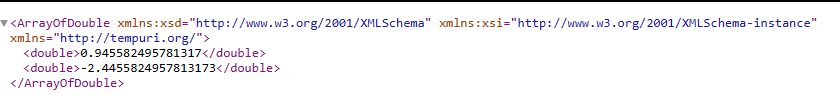
{

throw new NotImplementedException();

} } }







using Quadratic.localhost;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Quadratic

{

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

{

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

{

}

protected void btnSolve\_Click(object sender, EventArgs e)

{

localhost.WebService1 client = new localhost.WebService1();

double a = double.Parse(txtA.Text);

double b = double.Parse(txtB.Text);

double c = double.Parse(txtC.Text);

string method = ddlMethod.SelectedValue;

double[] solutions = null;

switch (method)

{

case "SolveQuadratic":

solutions = client.SolveQuadratic(a, b, c);

break;

case "SolveQuadraticByFactoring":

solutions = client.SolveQuadraticByFactoring(a, b, c);

break;

case "SolveQuadraticByCompletingTheSquare":

solutions = client.SolveQuadraticByCompletingTheSquare(a, b, c);

break;

default:

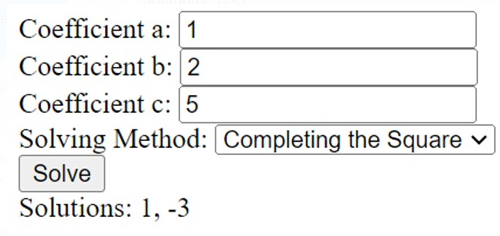
lblSolution.Text = "Invalid solving method selected.";

return;

}

lblSolution.Text = "Solutions: " + string.Join(", ", solutions);

}}}

A screenshot of a computer

Description automatically generated

A screenshot of a math form

Description automatically generated