ASP.NET Web Forms: Introduction

ASP.NET Web Forms is a web development framework provided by Microsoft. It allows developers to build dynamic, interactive web applications using server-side code. Web Forms are based on the Model-View-Controller (MVC) design pattern, which separates the application's logic, presentation, and data access layers.

Why Learn ASP.NET Web Forms in 2024?

Despite the rise of newer web development frameworks, ASP.NET Web Forms remains a popular choice for many developers, especially those who are familiar with the .NET platform. Here are a few reasons why you should consider learning ASP.NET Web Forms in 2024:

- **Established and Mature Framework:** ASP.NET Web Forms has been around for over two decades and has a large community of developers and resources. This means you can easily find support and guidance when learning and using the framework.
- **Ease of Use:** ASP.NET Web Forms is known for its user-friendly development environment and drag-and-drop interface. This makes it easier for beginners to learn and build web applications without writing complex code.
- **Integration with .NET:** ASP.NET Web Forms is tightly integrated with the .NET platform, which provides a wide range of libraries and tools for building robust and scalable web applications.
- **Enterprise-Level Support:** ASP.NET Web Forms is widely used in enterprise-level applications due to its reliability, scalability, and security features. Learning ASP.NET Web Forms can open up opportunities for you to work on large-scale projects.
- **Legacy Applications:** Many existing web applications are built using ASP.NET Web Forms, so learning the framework can be beneficial for maintaining and updating these applications.

Overall, ASP.NET Web Forms remains a valuable skill for students interested in web development, especially those targeting the .NET platform or enterprise-level applications. While newer frameworks may offer additional features and flexibility, ASP.NET Web Forms provides a solid foundation and a proven track record of success.

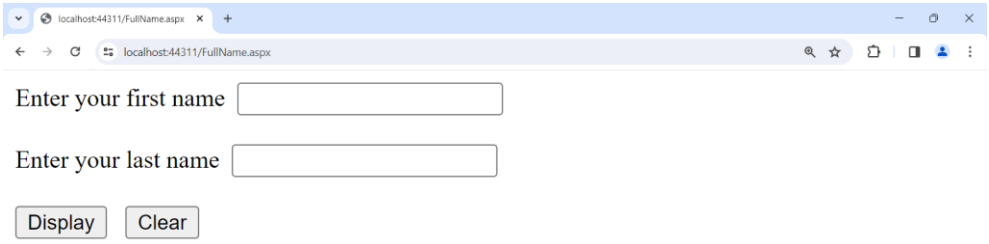
Task 1: Creating Your First ASP.NET Web Application

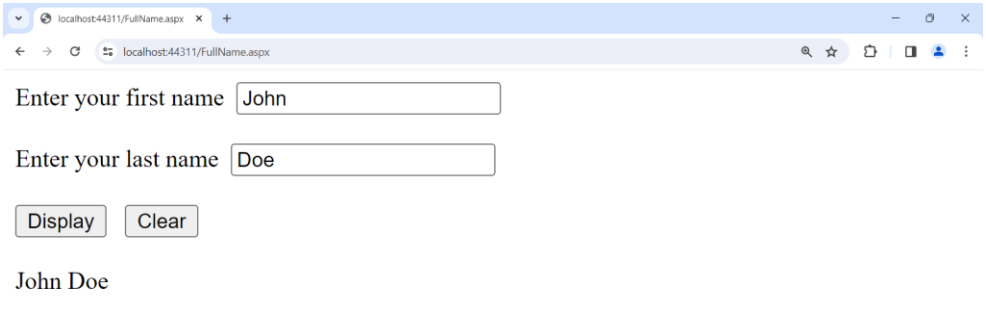
Objective:

- Create a web application using ASP.NET WebForms.
- Design the page with two textboxes (for first and last names), a button to display the full name, and a button to clear output.
- Display output using a label.

Instructions:

1. Create a new ASP.NET Web Forms project.
2. Design “FullName.aspx” with two textboxes, a "Show Name" button, a "Clear" button, and a label.
3. Implement code to concatenate first and last names and display the full name on button click.
4. Implement code to clear textboxes and label on "Clear" button click.





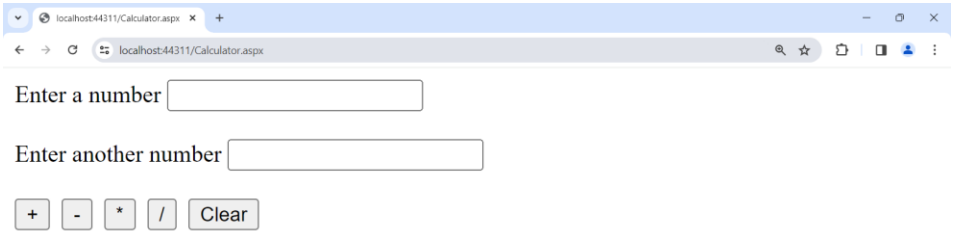
Task 2: Building a Calculator

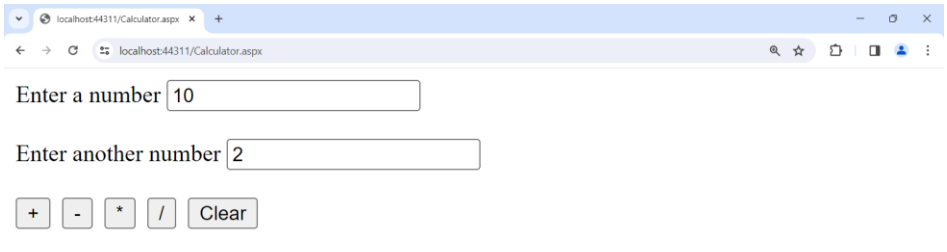
Objective:

- Create a calculator application using ASP.NET WebForms.
- Take two numbers and perform arithmetic operations.
- Display the result and include a clear button.

Instructions:

1. Design Calculator.aspx with textboxes for numbers, operations buttons (+, -, *, /), and a result label.
2. Implement event handlers for addition, subtraction, multiplication, and division.
3. Display the result on the label.
4. Include a button to clear the result field.





Task 3: Learning MasterPage, DropDownList, RadioButtonList, CheckBox, and QueryStrings

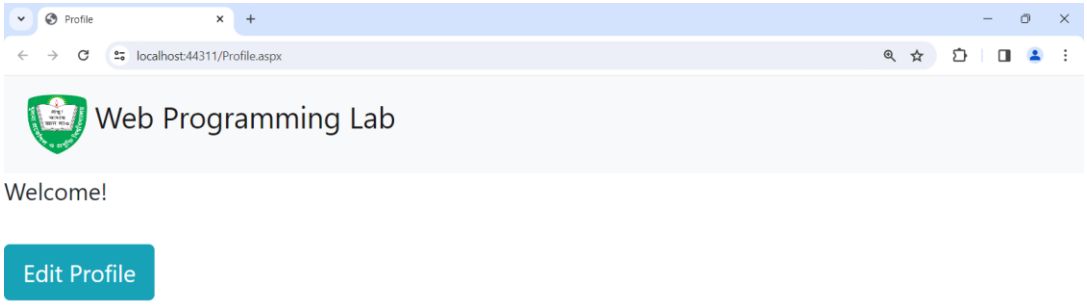
Objective:

- Learn about master pages, dropdown lists, radio button lists, check box, and query strings.
- Use master pages as containers and query strings to pass data between webforms.

Instructions:

1. Create a master page named "Site.master" with header, footer, and content placeholders.
2. Design "EditProfile.aspx" and "Profile.aspx" using the master page.
3. Add a dropdown list for occupation selection and a radio button list for gender in "EditProfile.aspx".
4. Design a "”Profile” class with the properties – “Name”, “Age”, “Occupation”, “Gender”; and methods – constructor, “GetProfileInfo()”.
5. Create an instance of “Profile” class in “EditProfile.aspx.cs” and if “Send to profile page” Checkbox is clicked, send the profile information from “EditProfile.aspx” to “Profile.aspx” using query strings.
6. Display user profile information in "Profile.aspx".
7. Test the functionality by navigating between pages and verifying data transfer.

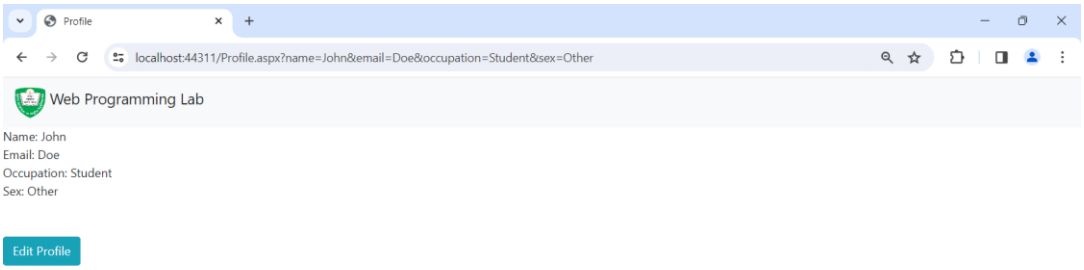
“Profile.aspx” in the beginning:



“EditPrfile.aspx”:

When “Send to profile page” is not checked:

When "Send to profile page" is checked, redirect to "Profile.aspx" and send data using query strings:



In Lab 5, you will learn about create, read, update, and delete (CRUD) database operations in ASP.NET WebForms using Microsoft SSMS. Alongside, you will also learn about GridView, PostBack, Session, and Cookies. You will find the codes of Lab 5 in these links: https://github.com/FarhanSadaf/WebForms-MSSQL-Create-Read-Functionality, https://github.com/FarhanSadaf/WebForms-MSSQL-Update-Delete-Functionality.

Completing these tasks will provide practical experience in ASP.NET WebForms and enhance understanding of key web programming concepts. Good luck!