Validation using JS:

**13) Develop a feature-rich ASP.NET web application that includes**

**various validation techniques and utilizes rich controls. The**

**application should have the following functionalities:**

**Form Validation:**

**a) Create a web page with a form that collects user registration**

**information (e.g., username, email, password).**

**b) Implement client-side validation using JavaScript/jQuery to**

**ensure that required fields are filled and validate the email**

**format.**

**c) Use ASP.NET validation controls to enforce server-side**

**validation for password complexity**

1)add webform:

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Validation.aspx.cs" Inherits="Validation\_JS\_Jquery.Validation" %>

<!DOCTYPE html>

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

<head runat="server">

<style>

.error-message {

color: red;

}

</style>

<title>Registration Form</title>

<script src="https://code.jquery.com/jquery-3.6.4.min.js"></script>

<script type="text/javascript">

$(document).ready(function () {

// Client-side validation using JavaScript/jQuery

$("#btnSubmit").click(function () {

var username = $("#txtUsername").val();

var email = $("#txtEmail").val();

var password = $("#txtPassword").val();

// Reset previous error messages

$(".error-message").text("");

// Validate required fields

if (username.trim() === '') {

$("#lblUsernameError").text("Username is required.");

return false;

}

if (email.trim() === '') {

$("#lblEmailError").text("Email is required.");

return false;

}

// Validate email format

var emailRegex = /^[^\s@]+@[^\s@]+\.[^\s@]+$/;

if (!emailRegex.test(email)) {

$("#lblEmailError").text("Invalid email format.");

return false;

}

if (password.trim() === '') {

$("#lblPasswordError").text("Password is required.");

return false;

}

return true;

});

});

</script>

</head>

<body>

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

<div>

<h2>User Registration</h2>

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

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

<asp:Label ID="lblUsernameError" runat="server" CssClass="error-message" Text=""></asp:Label>

<br /><br />

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

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

<asp:Label ID="lblEmailError" runat="server" CssClass="error-message" Text=""></asp:Label>

<br /><br />

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

<asp:TextBox ID="txtPassword" runat="server" TextMode="Password"></asp:TextBox>

<asp:Label ID="lblPasswordError" runat="server" CssClass="error-message" Text=""></asp:Label>

<br />

<asp:RegularExpressionValidator ID="revPassword" runat="server" ControlToValidate="txtPassword"

ErrorMessage="Password must be at least 8 characters long and contain at least one uppercase letter, one lowercase letter, and one number."

Display="Dynamic" ForeColor="Red" ValidationExpression="^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d).{8,}$">

</asp:RegularExpressionValidator>

<br /><br />

<asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit\_Click" />

</div>

</form>

</body>

</html>

2) **Webform1.aspx->webform1.aspx.cs=**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Validation\_JS\_Jquery

{

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

{

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

{

}

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

{

// Server-side validation using ASP.NET validation controls

if (Page.IsValid)

{

// Perform actions when the form is valid

string username = txtUsername.Text;

string email = txtEmail.Text;

string password = txtPassword.Text;

// Add your logic to save the data or perform other actions

}

}

}

}