

IN FOCUS



ASK A QUESTION

CONTRIBUTE

ARTICLE

READER LEVEL:



ASP.Net Form Validation Using JavaScript

By **Vithal Wadje** on Aug 15, 2013

In this article we will learn how to validate ASP.Net form controls using JavaScript.

411k

37

9

Download Files:

[Download 100% FREE Spire Office APIs](#)[FormValidationUsingJavaScript.zip](#)

Background

When you create an ASP.Net form then before submitting the form data on the server its necessary to ensure that the user has provided valid data to avoid any erroneous data to be inserted into the database. Their are many ways to validate ASP.Net form data, we will learn one by one. In this article we will learn about client-side validation that is done using JavaScript. So let us learn some basics because I have written this article only focusing on beginners and students.

What Form Validation is

Ensuring that data entered by the user into the form is proper and valid as per our business requirements is called as Form Validation.

Types of validation

- Client Side Validation
- Server Side Validation

Client side validation

Validation done in the browser before sending the form data to the server using JavaScript, jQuery and VBScript is called client-side validation.

Server Side Validation

Validation done at the server level after sending the form data to the server but before entering the data into the database is called server-side validation.

Where to write JavaScript code in ASP.Net Form ?

You can write the JavaScript code in the ASP.Net Page in the following sections.

- **Head Section**

You can write the JavaScript code in the head section, it is the recommended way to write the code, the code must be enclosed in the following syntax:

```
<head id="Head1" runat="server">
<script type="text/javascript">
    //write JavaScript code here
</script>
</head>
```

- **Body Section**

You can also write the JavaScript at the body section of the page, the function written inside the body tag will automatically be called after page load.

- **External file**

You can write the code by adding the JavaScript file template that is provided by the .Net framework and after that we can add the reference of the .js file in the head section or body section.

Creating the function

The function is created in JavaScript using the function keyword followed by name.

Syntax

```
function VildateData()
{
    //write code here
}
```

In the preceding syntax function is the keyword provided by the JavaScript to declare a function and the VildateData() is the function name, now write the code inside the function as in the following.

Example

```
function VildateData()
{
    alert("this is the JavaScript");
}
```

I hope you now understand the basics of validation in JavaScript, now let us create the one sample web application that demonstrates how to do validation.

Let us first create the web application with two web pages as in the following:

1. "Start" - "All Programs" - "Microsoft Visual Studio 2010"
2. "File" - "New Website" - "C# - Empty website" (to avoid adding a master page)

3. Give the web site a name, such as Validation or whatever you wish and specify the location
4. Then right-click on the solution in the Solution Explorer then select "Add New Item" - "Default.aspx page" (add two pages).

We are adding two web pages because our requirement is, in the first web page there is form data to be filled in by the user and only after validating the form data, the form will be redirected to the next page.

The first page source code <body> tag will look as in the following:

```
<body bgcolor="#3366ff">
  <form id="form2" runat="server">
    <br />
    <br />
    <div>
      <table>
        <tr>
          <td>
            Name
          </td>
          <td>
            <asp:TextBox ID="txtUserId" runat="server"></asp:TextBox>
          </td>
        </tr>
        <tr>
          <td>
            Email Id
          </td>
          <td>
            <asp:TextBox ID="txtmail" runat="server"></asp:TextBox>
          </td>
        </tr>
        <tr>
          <td>
            Gender
          </td>
          <td>
            <asp:DropDownList ID="ddlType" runat="server">
              <asp:ListItem Value="0">-Select-</asp:ListItem>
              <asp:ListItem Value="1">Male</asp:ListItem>
              <asp:ListItem Value="2">Female</asp:ListItem>
            </asp:DropDownList>
          </td>
        </tr>
        <tr>
          <td>
            word
          </td>
          <td>
            <asp:TextBox ID="txt1" runat="server" TextMode="word"></asp:TextBox>
          </td>
        </tr>
        <tr>
          <td>
            Confirm word
          </td>
          <td>
            <asp:TextBox ID="txt2" runat="server" TextMode="word"></asp:TextBox>
          </td>
        </tr>
      </table>
    </div>
  </form>
</body>
```

```

<tr>
  <td>
  </td>
  <td>
    <asp:Button ID="btnSave" runat="server" Text="Create" OnClientClick="return userValid();" />
    <asp:Button ID="Button1" runat="server" Text="Reset" />
  </td>
</tr>
</table>
<asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
</div>
</form>
</body>

```

Look at the preceding source code closely; see the id's of controls that play the important role in JavaScript validation by reading the ASP.Net control values in JavaScript code.

The design view of the preceding source code will look as in the following:

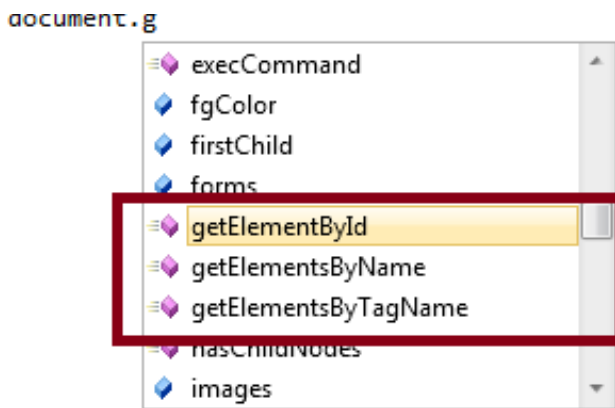
The screenshot shows the design view of a web page titled 'UserCreation.aspx'. The form is contained within a table. The table has five rows for input fields: 'Name', 'Email Id', 'Gender' (with a dropdown menu), 'Password', and 'Confirm Password'. Below the table, there are two buttons labeled 'Create' and 'Reset'. At the bottom of the form, there is a single-line label.

I hope you have created the same form as above for demonstration purposes.

Methods to read ASP.Net controls values in JavaScript are as:

C# Corner Discontinues The Code Snippet Section

NEWS CHAPTERS CAREER JOBS IDEAS



There are three main methods shown in the above image in the red square to read the ASP.Net controls values; they are:

getElementById: this method is used to read the controls values by their ID.

getElementByName: this method is used to read the controls values by their Name.

getElementByTagName: this method is used to read the controls values by their TagName.

In this article we use the [getElementById](#) method to read the control's values and external .js file to write JavaScript code, so let us see step-by-step how to add the JavaScript file.

1. Right-click on Solution Explorer then select "Add New Item" then the in the script.js page rename the .js page as you wish, I have renamed it to UserValidation.js
2. Create the function inside the UserValidation.js file named userValid() as:

```
function userValid()
{
    //write code here
}
```

Now declare the variable inside the function using the var keyword to read the ASP.Net control values by their ids and assign the values to the declared variable..

```
function userValid()
{
    var Name, , gender, con, EmailId, emailExp;
    Name = document.getElementById("txtUserId").value;
    gender = document.getElementById("ddlType").value;
    = document.getElementById("txt1").value;
    con = document.getElementById("txt2").value;
    EmailId = document.getElementById("txtmail").value;
    emailExp = /^[a-zA-Z0-9_\.\\-]+@[a-zA-Z0-9_\.\\-]+([com\co\.in])$/; // to validate email id
}
```

In the code above I have taken the ASP.Net control's values in variables so it cannot be repeated again and again in our function the emailExp variable holds the pattern of the email id in the form of a regular expression.

Now add the condition to ensure that all controls have a value, if the values are not entered in the form control then it \ a message. The condition will be as follows:

```
if (Name == " " && gender == 0 && == " " && con == " " && EmailId == "")
{
    alert( "Enter All Fields");
    return false;
}
```

In the preceding condition, to ensure that the form control's values are blank the message **Enter All Fields** is shown to the user and finally we are returning false; that is very important.

Importance of returning false

It's very important to use the return false statement after the condition block that return false so if validation determines that the business requirements are not met then the form cannot be submitted. If you do not return false then the message will be displayed to the user that all fields are required but the form will be post back and it gives you the second page directly. Therefore the return false statement works similar to the Required Field validator of ASP.Net. I hope you understand the concept.

The condition that checks both text boxes for word are as in the following:

```
if ( != con)
{
alert( "word not match");
return false;
}
```

In the preceding condition, we are checking that the two texboxes have words, in other words the word and confirm word.

Condition to determine if the email address is valid:

```
if (EmailId != "")
{
if (!EmailId.match(emailExp))
{
alert( "Invalid Email Id");
return false;
}
}
```

In the preceding condition, first we ensure that the email id is not blank then we match the email id entered into the text box to the Regular Expression that is saved in the emailExp variable.

The entire function will be as follows:

```
function userValid() {
var Name, , gender, con, EmailId, emailExp;

Name = document.getElementById("txtUserId").value;
gender = document.getElementById("ddlType").value;
= document.getElementById("txt1").value;
con = document.getElementById("txt2").value;
EmailId = document.getElementById("txtmail").value;
emailExp = /^[a-zA-Z0-9_\.\\-]+\@((([a-zA-Z0-9\\-]+\.)+([com\co\.\in])+$); // to validate email id

if (Name == "" && gender == 0 && == "" && con == "" && EmailId == "") {
    alert("Enter All Fields");
    return false;
}
if (Name == "") {
    alert("Please Enter Login ID");
    return false;
}
if (gender == 0) {
    alert("Please Select gender");
    return false;
}
if ( == "")
{
```

```

    alert("Please Enter word");
    return false;
}
if ( != " && con == ")
{
    alert("Please Confirm word");
    return false;
}
if ( != con)
{
    alert("word not match");
    return false;
}
if (EmailId == "")
{
    alert("Email Id Is Required");
    return false;
}
if (EmailId != "")
{
    if (!EmailId.match(emailExp))
    {
        alert("Invalid Email Id");
        return false;
    }
}
return true;
}

```

- Adding the reference of external JavaScript file into the Head section of ASP.Net form

Just drag the .js file from the Solution Explorer to the Head section of the ASP.Net form that automatically adds the file reference path, it will look like as in the following:

```

<head id="Head1" runat="server">
<script src="UserValidation.js" type="text/javascript">
</script>
</head >

```

- Calling JavaScript function on ASP.Net Button

To call the JavaScript function on the ASP.Net button we need to call the function in the ASP.Net button's OnClientClick property.

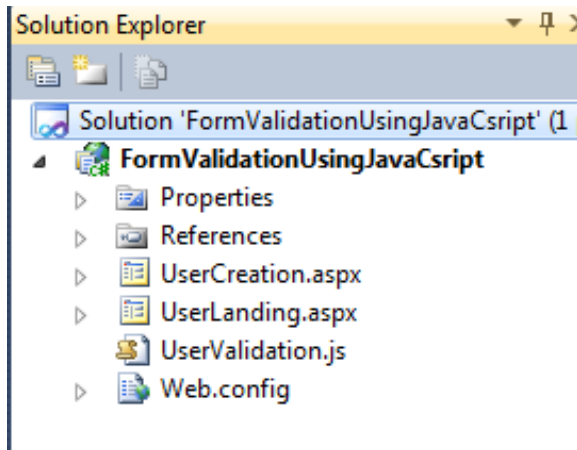
Example

```

<asp:Button ID="btnSave" runat="server" Text="Create" OnClientClick="return userValid();"/>

```

I hope you have understood it then the Solution Explorer will look like as in the following:



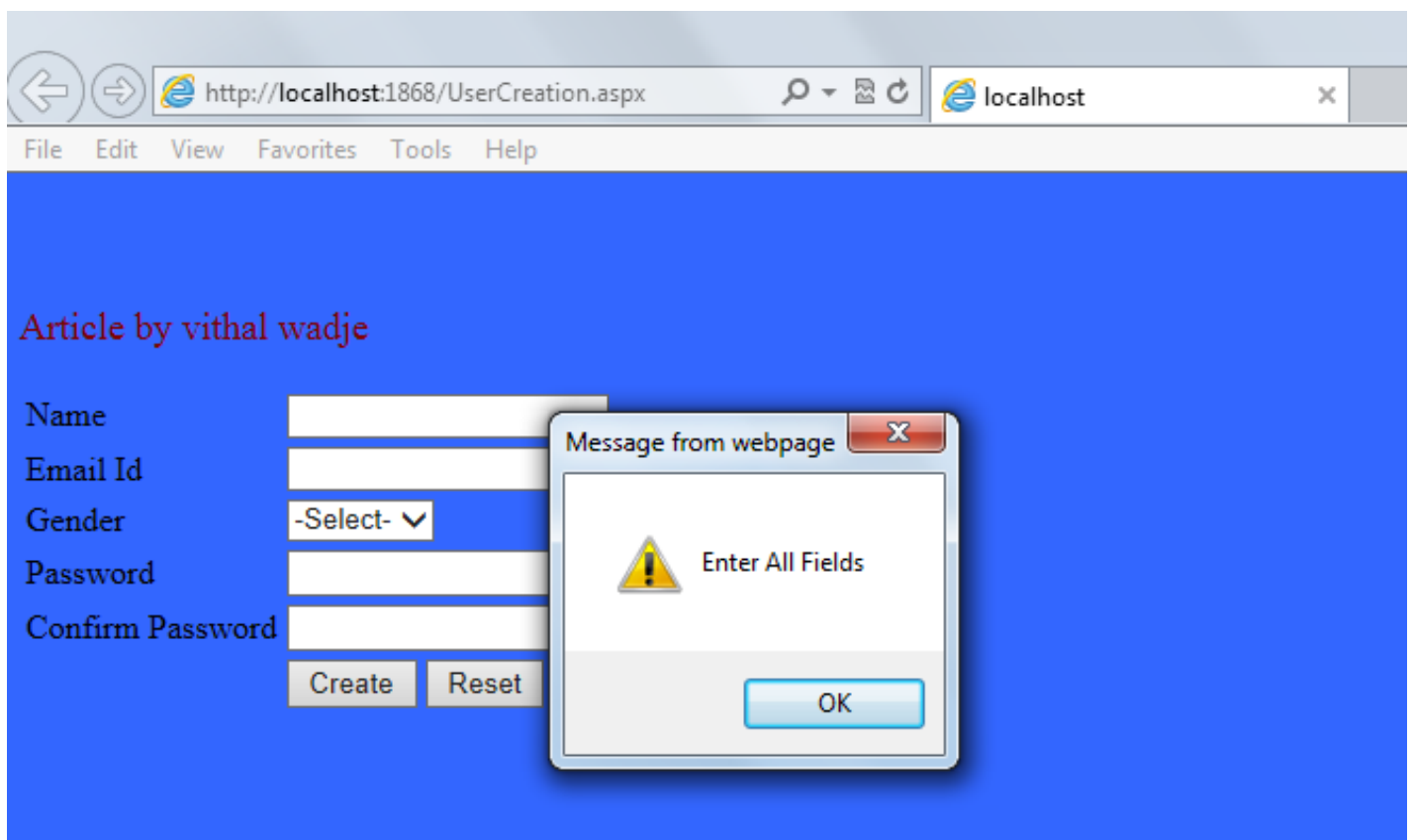
In the preceding Solution Explorer, there are two ASP.Net pages, UserCreation.aspx and UserLanding.aspx along with the UserValidation.js JavaScript file. In the UserCreation.aspx page the user enters the form details and then only after validating the details the page is redirected to the UserLanding.aspx page.

Use the following code in the create button:

```
protected void btn_Click(object sender, EventArgs e)
{
    Response.Redirect("UserLanding.aspx");
}
```

In the code above, only after validating the form data, the page is redirected to the UserLanding.aspx page.

Now run the ASP.Net web application and click on the Create button without inserting any data in the form, then it will show the following alert message.

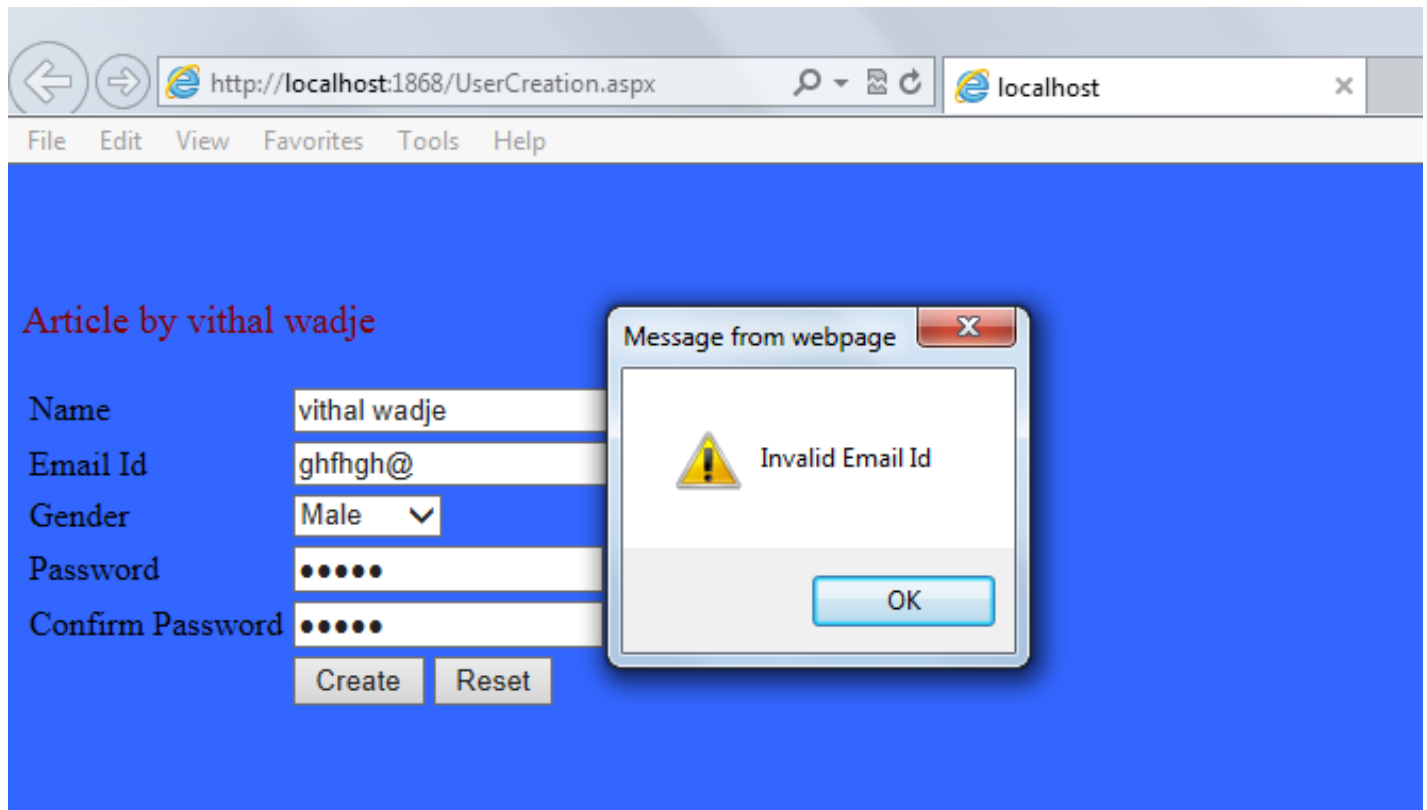


In the preceding screen you clearly see that even I have written the code on the create button to redirect to the next page but it will not be redirected because the form data is blank and it does not satisfy our validation condition that we set.

In other words, it's clear that the validation is done at the client side in the browser level and only validates the data; it will

execute the server-side code.

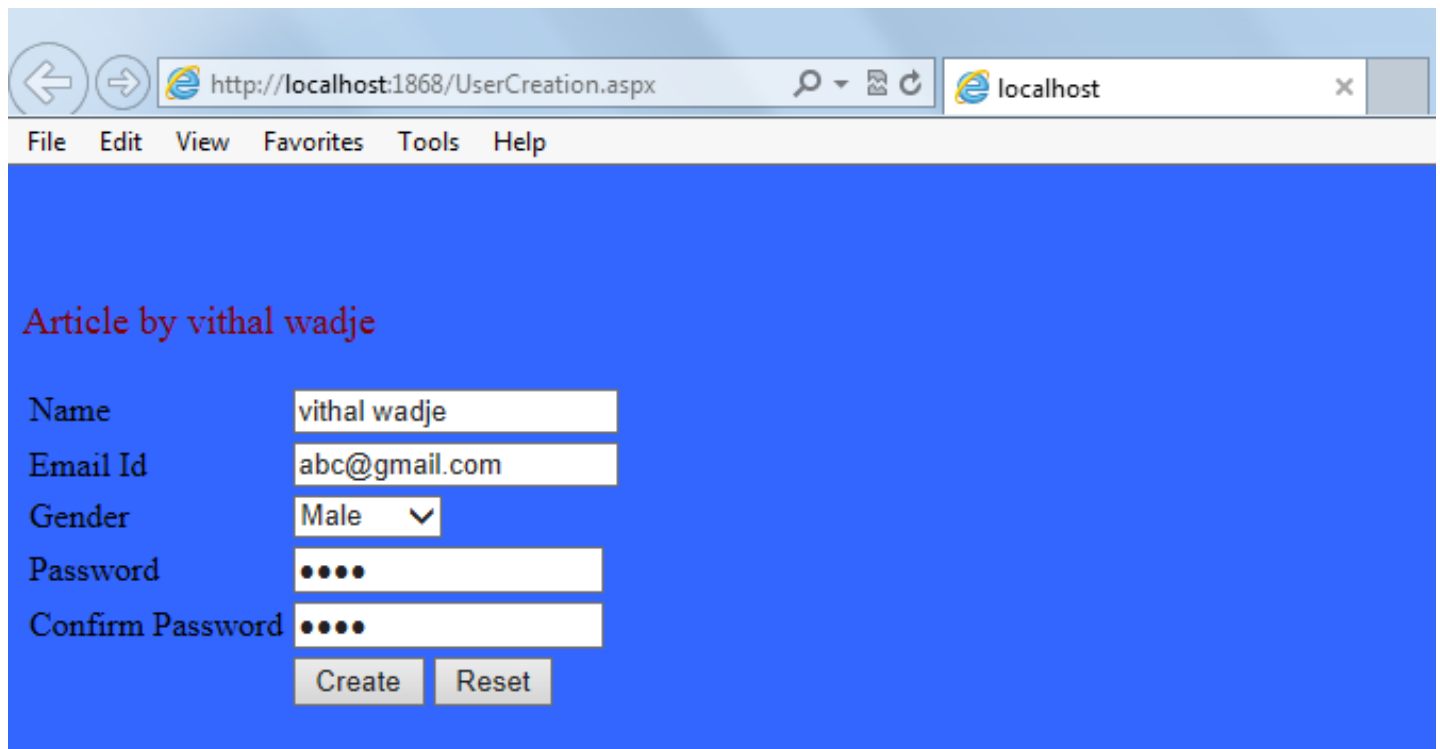
Now enter the invalid Email Id, it will show the following message:



Now enter the word that does not match the confirm word, it will show the following message:

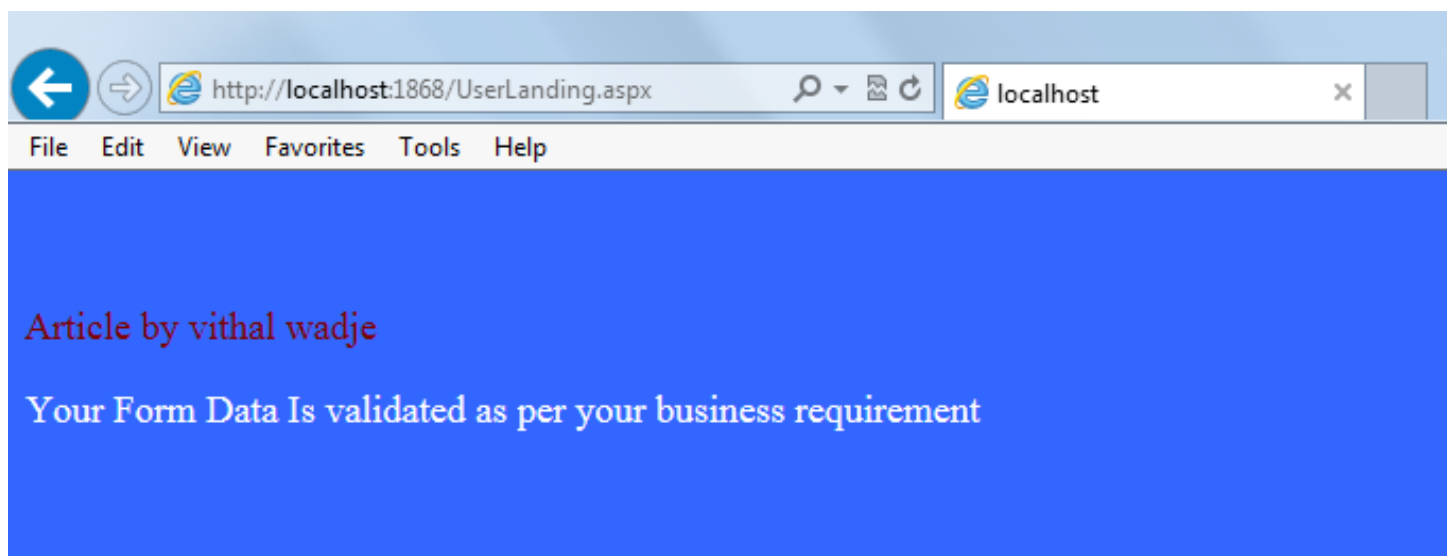


Now enter the valid details.



A screenshot of a web browser window. The address bar shows 'http://localhost:1868/UserCreation.aspx'. The browser has a menu bar with 'File', 'Edit', 'View', 'Favorites', 'Tools', and 'Help'. The page has a blue background. At the top, it says 'Article by vithal wadje' in red. Below that is a form with the following fields: 'Name' with the value 'vithal wadje', 'Email Id' with the value 'abc@gmail.com', 'Gender' with a dropdown menu showing 'Male', 'Password' with four dots, and 'Confirm Password' with four dots. At the bottom of the form are two buttons: 'Create' and 'Reset'.

Now click on the "Create" button; it will redirect to the next page as in the following:



A screenshot of a web browser window. The address bar shows 'http://localhost:1868/UserLanding.aspx'. The browser has a menu bar with 'File', 'Edit', 'View', 'Favorites', 'Tools', and 'Help'. The page has a blue background. At the top, it says 'Article by vithal wadje' in red. Below that, it says 'Your Form Data Is validated as per your business requirement' in white text.

Note

For detailed code please download the zip file attached above.

Summary

From all the examples above we see how to validate the form data using JavaScript. I hope this article is useful for all students and beginners. If you have any suggestion related to this article then please contact me.

[Download 100% FREE Spire Office APIs](#)

[Asp.net form](#) [client side validation](#) [java script](#) [JavaScript Validation](#)

**VITHAL WADJE** *TOP 50*

Vithal is a Developer , Software Architect, Microsoft MVP (2016) , C# Corner MVP (4 times) , Microsoft C# TechNet Guru (March 2015) , Author , Blogger ,Speaker , C# Corner Mumbai User Group Lead and Student's Mentor .He ... [Read more](#)

<http://www.compilemode.com>**15**
Rank**11.3m**
Read**Platinum**
Member**4**
Times**1**
Time

COMMENTS (37)

TRENDING UP

- 01 Using Generics With C#
- 02 When To Use Abstract Class and Interface In Real Projects
- 03 Web Crawling With C#
- 04 What Is The Future Of XAML
- 05 Best Practices For ASP.NET MVC Application
- 06 C# Version 7.0 Announcements
- 07 Introduction To .NET Core 1.0
- 08 Factory Design Pattern Real World Example
- 09 Highest Paying Tech Jobs Of 2016
- 10 Keep Learning, Or Become Dust!

[View All](#)

Follow @csharpcorner

75.1K followers

**C# Corner**

402,778 likes

Like Page**Share**

2 friends like this



[MVPs](#) [MOST VIEWED](#) [LEGENDS](#) [NOW](#) [PRIZES](#) [REVIEWS](#) [SURVEY](#) [DOWNLOADS](#)

Hosted By **CBeyond Cloud Services**

[ABOUT US](#) [FAQ](#) [MEDIA KIT](#) [MEMBERS](#) [STUDENTS](#) [LINKS](#)

[CONTACT US](#) [PRIVACY POLICY](#) [TERMS & CONDITIONS](#) [SITEMAP](#) [REPORT A BUG](#)

©2016 C# Corner. All contents are copyright of their authors.