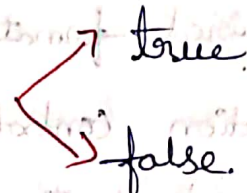


## \* Property With required field Validator

- Control to Validate
- Error message
- Set focus on error 
  - true
  - false
- Text
- Validation Group

## \* Control to Validate :-

This property is used to set id of the control for which we want to check user entered data or not.

## \* Error Message :-

This property is used to set or get the message we want to display to the user. When validation controls returns false.

## \* Set focus on error :-

When set to true → the focus will remain in the same control when required field validator returns false.

## \* Text

## \* Validation Group :-

Used to set or get the common group name for two or more controls for to which we would like to implement common validations.

→ Example to implement Required field Validator:-

Enter User Name  RfV1

Enter password

Login

lblDisplay

→ 2nd Web form you take "Welcome your login is success".

→ Create a New Website Add two Web pages with the Name sample1.aspx and sample2.aspx.

→ Then design both Web pages select RfV1 set the following properties.

→ Control to validate: txtUserName.

→ Error message: User Name is mandatory.

\* Write the following code for login button:-

```
using system.data;
using system.data.sql client;
namespace WebValidations
{
    public partial class sample1: System.Web.UI.Page
    {
        sql connection con; sql command cmd; sql dataReader
        DR;
        protected void Page_Load(object sender, event Args e)
        {
            string sql con string = "server = ";
            con = new sql connection (sql con string);
```

```

}
Protected Void btn_login_Click(Object Sender, EventArgs e)
{
    String Query = "Select * from users Where Username=@P1  

    and passWord=@P2";
    cmd = new SqlCommand(Query, con);
    cmd.CommandType = CommandType.Text;
    cmd.Parameters.AddWithValue("@P1", txtUserName.Text);
    cmd.Parameters.AddWithValue("@P2", txtPassWord.Text);
    con.Open();
    DR = cmd.ExecuteReader();
    if (DR.Read())
        Response.Redirect("Sample2.aspx");
    else
        lblDisplay.Text = "Invalid Credentials";
    con.Close();
}
}

```

⇒ Adding passWord is Mandatory :-

→ Create Another required field validation control

Rfv2 Set the following property.

1. Control to validate : txtPassWord
2. Error's message : passWord is Mandatory.



## 2. Regular Expression Validator:

This control is used to check whether user entered data is within the required format or not like email id format or pin-code format or internet address format etc.

### Properties With Regular Expression Validator:

1. Control to validate

2. Error message

3. Set focus on error

True

False

4. Text

5. Validation Group

6. Validation Expression

### \* Validation Expression:

→ This property is used to set or get the required expression format with which user entered data is to be compared.

→ If user entered data in control to validate matches with the format set in Validation Expression property then <sup>regular expression</sup> ~~required~~ ~~field~~ validator will return true. Otherwise ~~required~~ regular expression will return false.

→ Create a NEW Web page With the Name Sample3.aspx

Enter user Name

Enter password

→ Set the following properties for regular Expression Validator.

(i) Control to Validate : txtUserName

(ii) Error message : Invalid username format

\* Write the code for Sample1.aspx

\* Run the application And check.

→ In the above example if user not enter's user's Name and click on submit button then request to will be submitted to the server.

→ Here, Regular Expression Validator Will not check Whether user Entered data or not.

→ Except required field Validator no other Validation controls Will check Whether user Entered data or not.

→ To overcome this add required field Validator in the above Example.

→ To Work With Validations controls perform the following steps.

→ Go to Solution Explorer double click on Web.config file add the following code.

<app settings >

<add key="Validation Settings : UnObtrusive

Validation Mode" value="none">

</app settings >

### 3. Compare Validator:-

This control is used to compare the data of one control with the data of other control or with fixed value. if user entered data in control to validate matches with the data of control to compare or with value to compare property. Then compare validator will return true otherwise compare validator will return false.

### \* Properties With Compare Validator:-

1. Control to Validate
2. Error message
3. Set focus on Error 

True

false
4. Text
5. Validation Group
6. Control to compare
7. Type 

string

Integer

date

double

currency
8. Value to compare



## Control to Compare:-

→ used to set or get the id of the control

Whose data is compared with the data of control to validate.

→ if the data present in control to validate matches with the data of control to compare. Then compare validate will return true otherwise compare validate will return false.

## Type:-

→ used to set or get the required data type value with which the data is to be compared.

## Value to compare:-

→ used to set or get the fixed value with which data in control to validate is to be compared.

## \* Example to implement Compare Validator:-

Enter <sup>First</sup> Name

Enter Last Name

Enter username

Enter password

Retype password

Password does not match  
Password is mandatory.

Select Gender

• Male • female

Select Qualification

Select Skill

C#

C++

Java

Python

lblDisplay

Insert

→ Create a New Webpage design the Webpage Set the following properties

(i) Required field Validator :-

→ Control to Validate : txt password

→ Error Message : Password is Mandatory

(ii) Compare Validator :-

→ Control to Validate : txt password

→ Control to Compare : txt Retype password

→ Error message : passwords do not Match.

→ Write the following code :-

```
Protected void btnInsert_Click (object sender, EventArgs e)
{
    lblDisplay.Text = "User Created";
}
```