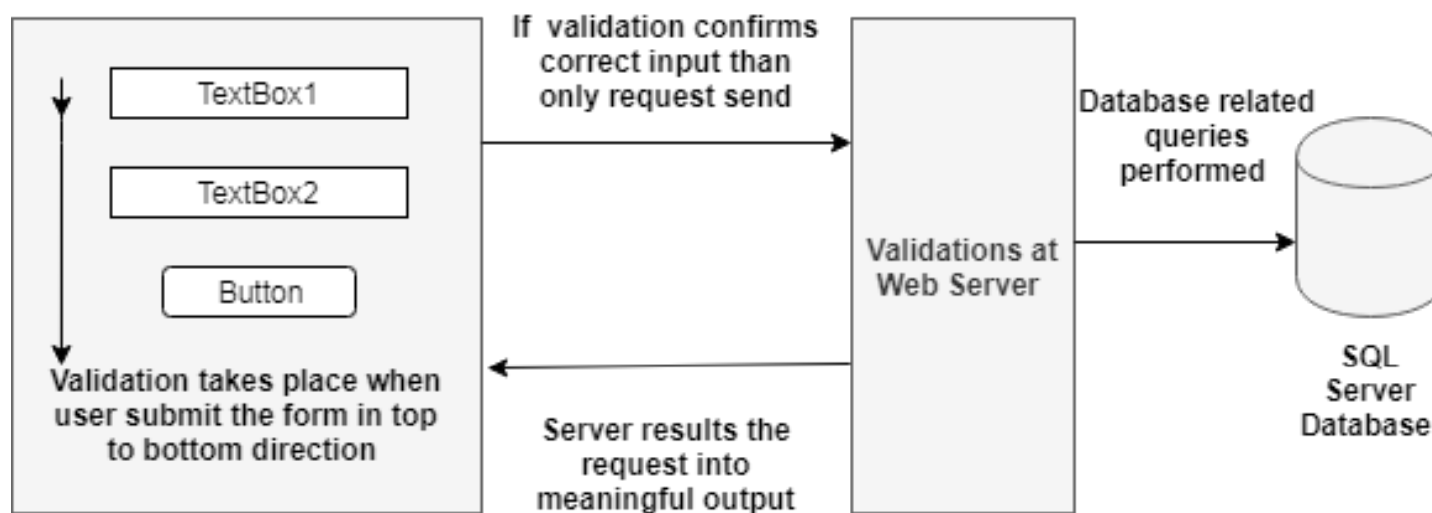


Name of the Teacher – Vaishali C. Dalvi
Class – B.Sc (CS) Entire
Subject – C# Programming

Validation Server Controls In ASP.Net

- Validation is important part of any web application.
- User's input must always be validated before sending across different layers of the application.
- **Validation controls are used to,**
 - Implement presentation logic.
 - To validate user input data.
 - Data format, data type and data range is used for validation.



Validation Process in ASP.NET Web Forms

- Validation can be performed at **the client-side** as well as **server-side**.
- Client-side validation makes the process **fast** as there is less number of hits to the server.
- Server-side validation is used to remove the limitation of client browsers dependencies and scripting language support.

Client Side Validation

- Client side validation is good but we have to be dependent on browser and scripting language support.
- Client side validation is considered convenient for users as they get instant feedback.
- The main advantage is that it prevents a page from being postback to the server until the client validation is executed successfully.
- For client script .NET used JavaScript. WebUIValidation.js file is used for client validation by .NET

Serve Side Validation

- For developer point of view serve side is preferable because it will not fail, it is not dependent on browser and scripting language.
- You can use ASP.NET validation, which will ensure client, and server validation.
- It work on both end; first it will work on client validation and then on server validation.
- At any cost server validation will work always whether client validation is executed or not. So you have a safety of validation check.

ASP.NET validation controls are derived from the abstract class i.e., **BaseValidator** Class, which is derived from **Label** Control class. The important properties and methods of the BaseValidator class are

- **ControlToValidate** – It indicates the input control to validate. It must be a unique value throughout the form. This attribute is a mandatory one as it is used to associate the input control with a validation control.
- **Enabled** – It enables or disables the validator.
- **Text** – It holds the message to be displayed in the event of a validation failure.
- **ErrorMessage** – The value in this attribute is displayed either when ValidationSummary control is used or when Text property is missing.
- **IsValid** – A Boolean attribute which indicates whether the control is valid or not.
- **Validate()** – This method revalidates the control and updates the IsValid

- Property named **ControlToValidate** cannot be left blank.
- Multiple validation controls can be assigned to a webform input control.
- **CauseValidation** property of the control must be “**true**” only then server validation will take place.
- By default, the validation controls perform validation on both the client (the browser) and the server.
- These validation controls uses JavaScript to perform validation on client-side.
- Client side validation is very fast because user gets immediate response whenever user enters an invalid value into a form field. Sometimes browsers (mobile browsers) do not support javascript.
- If a browser does not support JavaScript, then server side validation still perform.

Validation Controls in ASP.NET

- ASP.Net provides various validation controls that validate the user data to ensure that the data entered by the user are satisfied with the condition.
- **There are six types of validation controls in ASP.NET**
- RequiredFieldValidation Control
- CompareValidator Control
- RangeValidator Control
- RegularExpressionValidator Control
- CustomValidator Control
- ValidationSummary

RequiredFieldValidator Control

- The RequiredFieldValidator control ensures that the required field is not empty.
- It is generally tied to a text box to force input into the text box.
- It has certain properties that user need to set, for example: it should show an error message whenever the details are not filled in.
- **E.g.** For a login form user must enter the user name and password, both fields, to be filled in; only then can the user click the button and the server-side processing can occur. If either of the fields are empty then an error message should be invoked or displayed.

CompareValidator Control

- Used to perform three different types of validation tasks.
 - To perform a data type check.
 - To compare the value entered into a form field against a given fixed value.
 - To compare the value of one form field against another field.
- It has certain properties that user need to set, for example: it should mention controls to compare, which value to compare, and which operation to perform e.g. Equal, not equal to.
- **E.g.** For student data for calculating age, birth date entered must be less than current date.

RangeValidator Control

- The RangeValidator control simply specifies the permitted range within which the input value should fall.
- This is most helpful for numeral input values such as age or for Date input values.
- It has certain properties that user need to set, for example:
 - it should Specifies the type of data in the input field
The value is converted to this type before validation.
 - Minimum and maximum value that field can hold.
- **E.g.** For employee data age should be between 18 to 60.

RegularExpressionValidator Control

- RegularExpressions, or Regex, are patterns that define the format of the text.
- If the text is in the same format, Regex returns true, else false.
- A RegularExpressionValidator is a very versatile validation control. It matches the input text against the pattern specified in the ValidationExpression property.
- **E.g.** For entering valid email-id

CustomValidator Control

- [ASP.Net allows](#) the freedom of writing your own validator.
- This eases the task of a developer to validate the form at the client side itself. It also allows putting more complex validations in place. Validations that are business or application-specific can be written using custom validators.
- The custom validation code is written in a function in the code-behind page and the function name is passed as an attribute to the CustomValidator class. Custom validation can be done either at the client-side or the server-side.
- ClientValidationFunction property specifies that the validation is to be performed on the client-side. Such validation code must be written in some scripting language such as JavaScript, VBScript, etc.
- The ServerValidate event handler is used when validation has to be done on the server-side. The server-side validation routine is written in C#, VB .Net or any other .Net language.

ValidationSummary Control

- The ValidationSummary control does not perform any validation. Its purpose is to display a summary of all the errors on the page.

THANK

YOU