Validators in Web forms

Validators in web forms are controls, that validate a user input based on a condition.

Multiple validators can validate the same input control

Validators are asp.net controls and they add client (JS) validations automatically based on property settings

Validators can only validate asp.net controls (not html controls)

1. RequiredFieldValidator: Empty values are not allowed.
   Properties to set:
      ControlToValidate
      ErrorMessage
      SetFocusOnError
2. CompareValidator: Compares the values in 2 controls
   Properties to set:
      ControlToValidate
      ControlToCompare
      ErrorMessage
      SetFocusOnError
      Operator (optional. If not set, it will compare for equality)
      Type (optional. If not set, it will compare the values for string datatype)

3. RangeValidator
   Properties to set:
      ControlToValidate
      ErrorMessage
      SetFocusOnError
      MaximumValue
      MinimumValue
      Type (optional. If not set, it will assume string datatype)
4. RegularExpressionValidator
   Properties to set:
      ControlToValidate
      ErrorMessage
      SetFocusOnError
      ValidationExpression

5. CustomValidator: When none of the above fit the application requirement, custom validators are used
   Properties to set:
      ControlToValidate
      ErrorMessage
      SetFocusOnError
      ClientValidationFunction

==================================
Working with Asp.net Web Forms Application

Working with Database connection using Web Forms
Steps to insert or any activity with database
1. Connect to database.
   SqlConnection
2. Create a command to execute in the database
   SqlCommand
   SqlCommands example in SqlDataSource controls
   Please look at InsertCommand, UpdateCommand, DeleteCommand, SelectCommand
3. Every SqlCommand has parameters
   SqlParameter
4. Execute the command
   i. ExecuteNonQuery: Use for inserts. Gives how many records affected
   ii. ExecuteReader: Used to process record by record.. Connected Architecture
   iii. ExecuteScalar: Returns only one value
   iv. Fill: Return lot of data like a table

Page Lifecycle

Page_PreInit()
Page_Init()
Page_InitComplete()
Page_PreLoad()
Page_Load()
<Control Events>
Page_LoadComplete()
OnPreRender()
OnSaveStateComplete()
Render()

Page_Unload()

Reading Material:

https://www.tutorialspoint.com/asp.net/

Read more on State Management

https://www.tutorialspoint.com/asp.net/asp.net_managing_state.htm

Read more on Caching:

https://www.tutorialspoint.com/asp.net/asp.net_data_caching.htm

==

Summary for Partial View

1. Can render a partial view by returning a partial view

   Eg: return PatialView("pName")

2. Can render a partial view directly from razor code in .cshtml.

   Use the Html Helper

```
 @{
    Html.RenderPartial("PartialViewName", Model)
 }
```

==

Integration of Ajax with MVC

1. Use Nuget Packages. Install them for your MVC project
   a. Microsoft.jQuery.Unobtrusive.Ajax
   b. Microsoft.jQuery.Unobtrusive.Validation
2. Verify if configurations are added in Web.config.

   ** Collapse all the folders for your project. The last file in your project will have capital "W" in "Web.config"

   Check if the following is available inside <appSettings>

```
 <add key="ClientValidationEnabled" value="true" />
    <add key="UnobtrusiveJavaScriptEnabled" value="true" />
```

3. Open _Layout.cshtml. (Find this in Views => Shared => _Layout.cshtml)

   Add the following in the same sequence as mentioned

```
 <script src="~/Scripts/jquery-<version>.min.js"></script>
    <script src="~/Scripts/jquery.unobtrusive-ajax.min.js"></script>
```

   ** <version> : Look at the correct version of the file in the Scripts folder

4. Add an Action in the controller. The action should return a PartialView("yourView").

5. Create the Views for your action. Add @using(Ajax.BeginForm())

```
   Syntax for Ajax.BeginForm
    "YourActionNameOnClick", "ControllerName",
        new AjaxOptions
        {
            InsertionMode = InsertionMode.Replace | InsertionMode.InsertAfter |
InsertionMode.InsertBefore,
            HttpMethod = "Get | Post | Put | Delete",
            UpdateTargetId = "resultDiv"
        }
```

6. This completes the configuration for ajax using MVC, Ajax helpers