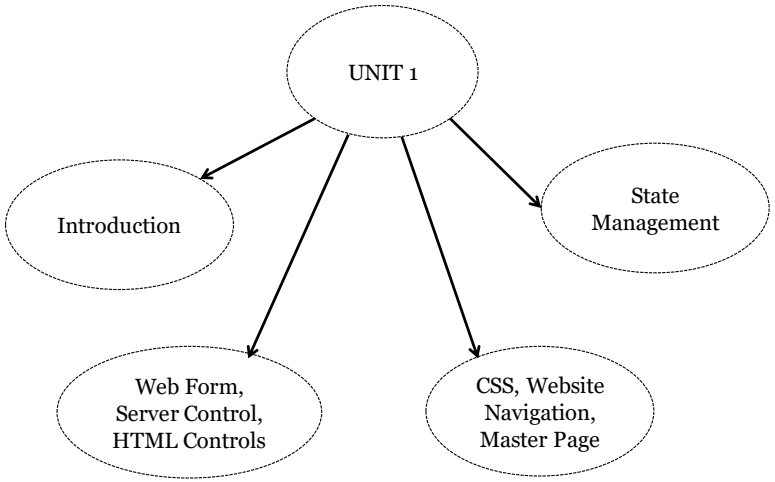
UNIT 1
Introduction of Web Development

CREATED BY ABHA DAMANI



1.1 Introduction

- How to create Project ?
- How to add new web form ?
- How to run project ?
- How to debugging ?
- How to add class ?

Website and Web Projects

- Project-based development
  - When you create a web project, Visual Studio generates a .csproj project file (assuming you're coding in C#) that records the files in your project and stores a few debugging settings. When you run a web project, Visual Studio compiles all your code into a single assembly before launching your web browser.
- Projectless development
  - Create a simple website without any project file. In this case, Visual Studio assumes that every file in the website directory (and its subdirectories) is part of your web application. In this scenario, Visual Studio doesn't need to precompile your code. Instead, ASP.NET compiles your website the first time you request a page. (Of course, you can use precompilation to remove the first-request overhead for a deployed web application. Chapter 18 explains how.)

Project Creation

- File
  - New
    - Project
      - Visual C#
        - Web
          - ASP.NET Empty Web Application
            - Name
              - ok.

Con.

- Server Explorer
- Toolbox
- Properties
- Solution Explorer
- Team Explorer
- Class View

Con..

- using System.Web.UI
  - The System.Web.UI namespace provides classes and interfaces that enable you to create ASP.NET server controls and ASP.NET Web pages for the user interface of your ASP.NET Web applications.
- using System.Web.UI.WebControls
  - The System.Web.UI.WebControls namespace contains classes that allow you to create Web server controls on a Web page. Web server controls run on the server and include form controls such as buttons and text boxes

Run Project

- Press F5.

Debugging

- Add Break Point.
  - Press Fn + F11 .

Add Class

- Select project folder from solution explorer.
  - Right click
    - Add
      - Class
        - Class name
          - Add
- Namespaces
  - using System.Linq
    - provides classes and interfaces that support queries that use Language-Integrated Query (LINQ).
  - using System.Collections.Generic
    - contains interfaces and classes that define generic collections, which allow users to create strongly typed collections that provide better type safety and performance than non-generic strongly typed collections.

Con..

- using System.Web
  - The System.Web namespaces contain types that enable browser/server communication.
- using System
  - The System namespace contains fundamental classes and base classes that define commonly-used value and reference data types, events and event handlers, interfaces, attributes, and processing exceptions.

1.2 Web Form, Server Control, HTML Controls

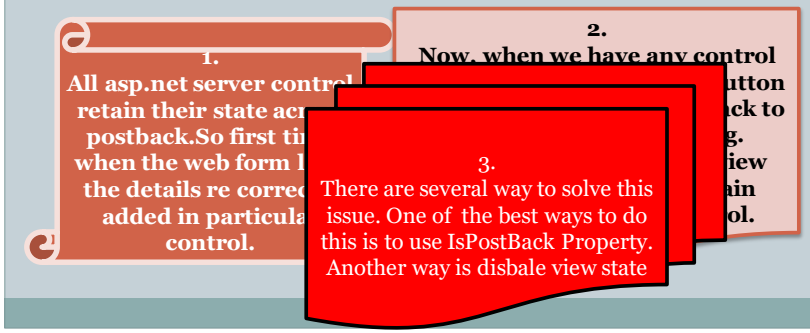
- Adding web form
- Server control
- Html control

Add New Form

- Select Project from Solution Explorer.
  - Right click
    - ADD
      - Web Form
        - Form name
          - Ok
- Design
- Split
- Source
- Output
- Data Tools Operations
- Error List

IsPostBack Property

- It is page level property, that can be used to determine whether the page is being loaded in response to a client post back, or if it is being loaded and accessed for the first time.



Server Control

- ASP.NET server controls are the primary controls used in ASP.NET. These controls can be grouped into the following categories:
  - **Validation controls** - These are used to validate user input and they work by running client-side script.
  - **Data source controls** - These controls provides data binding to different data sources.
  - **Data view controls** - These are various lists and tables, which can bind to data from data sources for displaying.
  - **Personalization controls** - These are used for personalization of a page according to the user preferences, based on user information.

Con..

- **Login and security controls** - These controls provide user authentication.
- **Master pages** - These controls provide consistent layout and interface throughout the application.
- **Navigation controls** - These controls help in navigation. For example, menus, tree view etc.
- **Rich controls** - These controls implement special features. For example, AdRotator, FileUpload, and Calendar control.

Property	Description
BackColor	Background color.
BorderColor	Border color.
BorderStyle	Border style.
BorderWidth	Border width.
CssClass	CSS class
Enabled	Indicates whether the control is grayed out.
Font	Font.
Forecolor	Foreground color.
Height	Height in pixels or %.
ID	Identifier for the control.
IsEnabled	Gets a value indicating whether the control is enabled.
TabIndex	Gets or sets the tab index of the Web server control.
ViewState	Gets a dictionary of state information that saves and restores the view state of a server control across multiple requests for the same page.
Visible	It indicates whether a server control is visible.
Width	Gets or sets the width of the Web server control.

Button Controls

- ASP.NET provides three types of button control:
- **Button** : It displays text within a rectangular area.
- **Link Button** : It displays text that looks like a hyperlink.
- **Image Button** : It displays an image.

```
<asp:Button ID="Button1"
runat="server"
onclick="Button1_Click" Text="Click"
/ >
```

Property	Description
Text	The text displayed on the button. This is for button and link button controls only.
ImageUrl	For image button control only. The image to be displayed for the button.
AlternateText	For image button control only. The text to be displayed if the browser cannot display the image.
CausesValidation	Determines whether page validation occurs when a user clicks the button. The default is true.
CommandName	A string value that is passed to the command event when a user clicks the button.
CommandArgument	A string value that is passed to the command event when a user clicks the button.
PostBackUrl	The URL of the page that is requested when the user clicks the button.

Text Boxes and Labels

- Text box controls are typically used to accept input from the user. A text box control can accept one or more lines of text depending upon the settings of the TextMode attribute.
- Label controls provide an easy way to display text which can be changed from one execution of a page to the next. If you want to display text that does not change, you use the literal text.

```
<asp:TextBox ID="txtstate"
runat="server"  ></asp:TextBox>
```

Property	Description
TextMode	Specifies the type of text box. SingleLine creates a standard text box, MultiLIne creates a text box that accepts more than one line of text and the Password causes the characters that are entered to be masked. The default is SingleLine.
Text	The text content of the text box.
MaxLength	The maximum number of characters that can be entered into the text box.
Wrap	It determines whether or not text wraps automatically for multi-line text box; default is true.
ReadOnly	Determines whether the user can change the text in the box; default is false, i.e., the user can not change the text.
Columns	The width of the text box in characters. The actual width is determined based on the font that is used for the text entry.
Rows	The height of a multi-line text box in lines. The default value is 0, means a single line text box.
ToolTip	It displayed when the mouse is over the control.
Height	Set the height
Width	Set the width

Check Boxes and Radio Buttons

- A check box displays a single option that the user can either check or uncheck and radio buttons present a group of options from which the user can select just one option.
- To create a group of radio buttons, you specify the same name for the GroupName attribute of each radio button in the group. If more than one group is required in a single form, then specify a different group name for each group.
- If you want check box or radio button to be selected when the form is initially displayed, set its Checked attribute to true. If the Checked attribute is set to true for multiple radio buttons in a group, then only the last one is considered as true.

```
<asp:CheckBox ID=
"chkoption" runat=
"Server">
</asp:CheckBox>
```

```
<asp:RadioButton ID=
"rdboption" runat=
"Server"> </asp:
RadioButton>
```

Property	Description
Text	The text displayed next to the check box or radio button.
Checked	Specifies whether it is selected or not, default is false.
GroupName	Name of the group the control belongs to.
TextAlign	On which side of the control the text should appear. Left or Right.

List Controls

- ASP.NET provides the following controls
  - Drop-down list,
  - List box,
  - Radio button list,
  - Check box list,
  - Bulleted list.
- These control let a user choose from one or more items from the list. List boxes and drop-down lists contain one or more list items. These lists can be loaded either by code or by the ListItemCollection editor.

```
<asp:ListBox
ID="ListBox1"
runat="server"
AutoPostBack="True"
OnSelectedIndexChanged=
"ListBox1_SelectedIndex
Changed">
</asp:ListBox>
```

```
<asp:DropDownList
ID="DropDownList1"
runat="server"
AutoPostBack="True"
OnSelectedIndexChanged="DropDow
nList1_SelectedIndexChanged">
</asp:DropDownList>
```

Property	Description
Items	The collection of ListItem objects that represents the items in the control. This property returns an object of type ListItemCollection.
Rows	Specifies the number of items displayed in the box. If actual list contains more rows than displayed then a scroll bar is added.
SelectedIndex	The index of the currently selected item. If more than one item is selected, then the index of the first selected item. If no item is selected, the value of this property is -1.
SelectedValue	The value of the currently selected item. If more than one item is selected, then the value of the first selected item. If no item is selected, the value of this property is an empty string ("").
SelectionMode	Indicates whether a list box allows single selections or multiple selections.

Common properties of each list item objects:

Property	Description
AlternateText	Alternate text to be displayed in absence of the image.
ImageAlign	Alignment options for the control.
ImageUrl	Path of the image to be displayed by the control.

Con..

HTML Control/Web Control

View State

- Web applications work on HTTP protocol. The HTTP protocol is stateless, meaning it does not retain state between user requests.
- It stores data in the hidden format.
- If you are using server control than internal server control stores its value in hidden format.
- It stores data in an encoded base 64 formats.
- Syntax for creating view state:-
  ViewState[“variablename”]= value
- Ex.Count=1;ViewState[“click”]= Count+1;

1.3. Add CSS

- Select project from Solution Explorer
  - Right click
    - Add
      - Style sheet
        - Name
          - Ohk

Apply CSS

- Internal CSS
- External CSS
- Inline CSS

1.5 State Management

- To manage the state of data we need to use a technique that provides functionality by which we can send data from one page to another page.
- There is the following technique to implement state management.
  - Query String
  - Cookies
  - Session State
  - Application State

Session State

- Session variables are available across all pages, but for a given single session only.
  - It is like single user global data.
  - Only current session has access to its session state.
  - It is a unique instance of a browser.
- We can create a cookie less session by setting parameter of session in global.asax file.
- <sessionstate mode="InProc" cookiless="true"/>

The HttpSessionState class has the following properties:

Properties	Description
SessionID	The unique session identifier.
Item(name)	The value of the session state item with the specified name. This is the default property of the HttpSessionState class.
Count	The number of items in the session state collection.
TimeOut	Gets and sets the amount of time, in minutes, allowed between requests before the session-state provider terminates the session.

The HttpSessionState class has the following methods:

Methods	Description
Add(name, value)	Adds an item to the session state collection.
Clear	Removes all the items from session state collection.
Remove(name)	Removes the specified item from the session state collection.
RemoveAll	Removes all keys and values from the session-state collection.
RemoveAt	Deletes an item at a specified index from the session-state collection.

Application State

- Application variables are available across all pages and across all session.
  - It is like multi user global data.
  - All session can read and write application session variable.
  - Syntax:-

Application[“variable name”]= value

  - Ex:-

Application[“count”]= no+1;

- Application state data is generally maintained by writing handlers for the events:
  - Application_Start
  - Application_End
  - Application_Error
  - Session_Start
  - Session_End

The HttpApplicationState class has the following properties:

Properties	Description
Item(name)	The value of the application state item with the specified name. This is the default property of the HttpApplicationState class.
Count	The number of items in the application state collection.

The HttpApplicationState class has the following methods:

Methods	Description
Add(name, value)	Adds an item to the application state collection.
Clear	Removes all the items from the application state collection.
Remove(name)	Removes the specified item from the application state collection.
RemoveAll	Removes all objects from an HttpApplicationState collection.
RemoveAt	Removes an HttpApplicationState object from a collection by index.
Lock()	Locks the application state collection so only the current user can access it.
Unlock()	Unlocks the application state collection so all the users can access it.

View State vs Session State vs Application State

View State	Session State	Application Session
view State of web form is available only with in that web form.	Session state variable are available across all pages, but only for a single session.	Application state variable are available across all pages and across all session.
-	It is like single user global data.	It is like multi user global data.
It stored on the page in a hidden field called _ViewState	It stored on the web server.	It stored on the web server.
It is used by all asp.net controls to retain their state across post back.	It is clear when the user session is timeout. Default timeout is 20 minutes. Configurable in web.config file	It cleared, when the process hosting system is restarted.
ViewState[“variablename”]=value	SessionState[“variablename”]=value	ApplicationState[“variablename”]=value