UNIT II: ASP.NET WITH C#

UNIVERSITY QUESTIONS WITH ANSWER

1. What is the difference between List Box and Drop-Down Lists? List and explain any three common properties of these controls.

List box:

The ListBox control enables you to display a list of items to the user that the user can select by clicking.

In addition to display and selection functionality, the ListBox also provides features that enable you to efficiently add items to the ListBox and to find text within the items of the list. You can use the Add or Insert method to add items to a list box.

Drop-down list:

Use the **DropDownList** control to create a single selection drop-down list control. You can control the appearance of the **DropDownList** control by setting the **BorderColor**, **BorderStyle**, and **BorderWidth** properties.

To specify the items that appear in the **DropDownList** control, place a **ListItem** element for each entry between the opening and closing tags of the **DropDownList** control.

The **DropDownList** control also supports data binding. To bind the control to a data source, first create a data source, such as a **System.Collections.ArrayList**, that contains the items to display in the control. The **DropDownList** control will now display the information from the data source.

Common Properties of List box and Drop-down Lists:

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
Text	The text displayed for the item
Selected	Indicates whether the item is selected.
Value	A string value associated with the item.

It is important to notes that:

- To work with the items in a drop-down list or list box, you use the Items property of the control. This property returns a ListItemCollection object which contains all the items of the list.
- The SelectedIndexChanged event is raised when the user selects a different item from a drop-down list or list box.

2. Explain AdRotator control with example in ASP.NET

- The AdRotator control randomly selects banner graphics from a list, which is specified in an external XML schedule file. This external XML schedule file is called the advertisement file.
- The AdRotator control allows you to specify the advertisement file and the type of window that the link should follow in the AdvertisementFile and the Target property respectively.
  The basic syntax of adding an AdRotator is as follows:

  <asp:AdRotator runat = "server" AdvertisementFile = "adfile.xml" Target = "_blank" />
  The advertisement file is an XML file, which contains the information about the advertisements to be displayed.

The Advertisement File

- The advertisement file is an XML file, which contains the information about the advertisements to be displayed.
- Extensible Markup Language (XML) is a W3C standard for text document markup. It is a text-based markup language that enables you to store data in a structured format by using meaningful tags. The term 'extensible' implies that you can extend your ability to describe a document by defining meaningful tags for the application.

Following is an example of XML file:

```
<BOOK>
  <NAME> Leaarn ASP </NAME>
  <AUTHOR> MOHADDESA ZAIDI </AUTHOR>
  <PUBLISHER> RIZVI COLLLEGE </PUBLISHER>
  <PRICE> 300.00</PRICE>
</BOOK>
```

There are the following standard XML elements that are commonly used in the advertisement file:

Element	Description
Advertisements	Encloses the advertisement file.
Ad	Delineates separate ad.
ImageUrl	The path of image that will be displayed.
NavigateUrl	The link that will be followed when the user clicks the ad.
AlternateText	The text that will be displayed instead of the picture if it cannot be displayed.
Keyword	Keyword identifying a group of advertisements. This is used for filtering.
Impressions	The number indicating how often an advertisement will appear.
Height	Height of the image to be displayed.
Width	Width of the image to be displayed

Example

```
<form id="form1" runat="server">
<div>
<asp:AdRotator ID="AdRotator1" runat="server" AdvertisementFile
="~/ads.xml" onadcreated="AdRotator1_AdCreated" />
</div>
</form>
```

3. List and explain any four types of validation controls used in ASP.NET

- ASP.NET validation controls validate the user input data to ensure that useless, unauthenticated, or contradictory data don't get stored.
- ASP.NET provides the following validation controls :
    1. RequiredFieldValidator
    2. RangeValidator
    3. CompareValidator
    4. RegularExpressionValidator
    5. CustomValidator
    6. ValidationSummary

➢ RequiredFieldValidator :

This validator is used to make an input control required. It will throw an error if user leaves input control empty.

It is used to mandate form control required and restrict the user to provide data.

EXAMPLE :

```
<asp:TextBox ID="username" runat="server">
</asp:TextBox>
<asp:RequiredFieldValidatorIDasp:RequiredFieldValidatorID="user" runat="
server" ControlToValidate="username"  ErrorMessage="Please enter a user name" For
eColor="Red">
</asp:RequiredFieldValidator>
```

➢ RangeValidator :

RangeValidator control is used to checks the input control's value is within a specified range, Means input values must be between two values, It has minimum and maximum value.

Generally we use RangeValidator control for Mobile no and Pincode Number.

The most important properties of the RangeValidator control are **MaximumValue**, **MinimumValue**, and **Type**.

EXAMPLE :

```
<asp:RangeValidator ID="RangeValidator1" runat="server"
 ControlToValidate="txtmobile" ErrorMessage="Mobile must be 10 digit !!"
MaximumValue="9999999999" MinimumValue="1000000000" Type="Double">
```

```
</asp:RangeValidator>
```

- **CompareValidator**

This validator evaluates the value of an input control against another input control on the basis of specified operator.

EXAMPLE :

```
< asp:CompareValidator ID="CompareValidator1" runat="server" ControlToCo
mpare="secondval"   ControlToValidate="firstval" Display="Dynamic" ErrorMes
sage="Enter valid value" ForeColor="Red"   Operator="LessThan" Type="Integer
">
</asp:CompareValidator>
```

- **ValidationSummary**

This validator is used to display list of all validation errors in the web form. It allows us to summarize the error messages at a single location.

EXAMPLE :

```
<asp:ValidationSummary ID="ValidationSummary1" runat="server" ForeColor
="Red"/>
```

4. **Explain Calendar control with example in ASP.NET**

SOLUTION :

- ASP.NET provides a Calendar control that is used to display a calendar on a Web page. ASP.NET Calendar control displays a month calendar that allows user to select dates and move to the next and previous months.
- By default, this control displays the name of the current month, day headings for the days of the weeks, days of the month and arrow characters for navigation to the previous or next month.
- The calendar control is a functionally rich web control, which provides the following capabilities:
    - Displaying one month at a time
    - Selecting a day, a week or a month
    - Selecting a range of days
    - Moving from month to month

- Controlling the display of the days programmatically

The basic syntax of a calendar control is:

<asp:Calender ID = "Calendar1" runat = "server"> </asp:Calender>

EXAMPLE:

```
<asp:Calendar ID="Calendar1" runat="server" BackColor="#FFFFCC" BorderColor="#FFCC66" BorderWidth="1px" DayNameFormat="Shortest" Font-Names="Verdana" FontSize="8pt" ForeColor="#663399" ShowGridLines="True" OnDayRender="Calendar1_DayRender" OnSelectionChanged="Calendar1_SelectionChanged"
  OnVisibleMonthChanged="Calendar1_VisibleMonthChanged">
<SelectedDayStyle BackColor="#CCCCFF" Font-Bold="True" />
<SelectorStyle BackColor="#FFCC66" />
<TodayDayStyle BackColor="#FFCC66" ForeColor="White" />
 <OtherMonthDayStyle ForeColor="#CC9966" />          <NextPrevStyle Font-Size="9pt" ForeColor="#FFFFCC" />
<DayHeaderStyle BackColor="#FFCC66" Font-Bold="True" Height="1px" />
<TitleStyle BackColor="#990000" Font-Bold="True" FontSize="9pt" ForeColor="#FFFFCC" />  </asp:Calendar>
```

Properties and Events of the Calendar Control

The calendar control has many properties and events, using which you can customize the actions and display of the control. The following table provides some important properties of the Calendar control:

Properties	Description
Caption	Gets or sets the caption for the calendar control.
CaptionAlign	Gets or sets the alignment for the caption.
CellPadding	Gets or sets the number of spaces between the data and the cell border.
CellSpacing	Gets or sets the space between cells.
DayHeaderStyle	Gets the style properties for the section that displays the day of the week.
DayNameFormat	Gets or sets format of days of the week.

DayStyle	Gets the style properties for the days in the displayed month.
FirstDayOfWeek	Gets or sets the day of week to display in the first column.
NextMonthText	Gets or sets the text for next month navigation control. The default value is >.
NextPrevFormat	Gets or sets the format of the next and previous month navigation control.

The Calendar control has the following three most important events that allow the developers to program the calendar control. They are:

Events	Description
SelectionChanged	It is raised when a day, a week or an entire month is selected.
DayRender	It is raised when each data cell of the calendar control is rendered.
VisibleMonthChanged	It is raised when user changes a month.

5. Short note on Page class.

When an ASP.NET page is requested and it automatically renders the markup code which creates an instance of a class that represents your paf\ge.
It combines both the aspx file and scripting code.
An ASP.NET page runs as a unit, combining the server-side elements in a page.

PROPERTIES:

- **IsPostBack:**
  - This ensures whether the page is executed for the first time. If it is for the first time it is false otherwise it will be true.
- **EnableViewState:**
  - If true we can store state information otherwise not possible.
- **Application:**
  - Collection of information shared among all the users about website.
    For example : Counting the number of visitors of a webpage.
- **Session:**
  - Information for a single user, so it can be used in different pages.

- **Cache:**
  - Store objects that are time-consuming to create so they can be reused in other pages or for other clients.
- **Request:**
  - The Http Response object that contains information about the current web request.
- **Response:**
  - The Http Response object that represents the response ASP.NET will send to the user's browser.
- **Server:**
  - The Http Server Utility object that allows to perform a few miscellaneous tasks.
- **User:**
  - To store the information of authenticated user.

6. Explain SiteMapPath control in ASP.NET

- The SiteMapPath control basically is used to access web pages of the website from one webpage to another. It is a navigation control and displays the map of the site related to its web pages.
- This map includes the pages in the particular website and displays the name of those pages. We can click on that particular page in the Site Map to navigate to that page. We can say that the SiteMapPath control displays links for connecting to URLs of other pages.
- The SiteMapPath control uses a property called SiteMapProvider for accessing data from databases and it stores the information in a data source. The SiteMapProvider internally utilizes the SiteMapProvider abstract class defined under the **System.Web** namespace.

Public Properties of SiteMapPath class :

- **ParentLevelsDisplayed:** It specifies the number of levels of parent nodes and then displays the control accordingly related to the currently displayed node.
- **RenderCurrentNodeAsLink:** It specifies whether or not the site navigation node that represents the currently displayed page is rendered as a hyperlink.
- **PathSeperator:** It specifies the string that displays the SiteMapPath nodes in the rendered navigation path.

EXAMPLE :

```
<asp:SiteMapDataSource ID="SiteMapDataSource1" runat="server" ShowStartingNode
="true" />

<asp:SiteMapPath ID="SiteMapPath1" runat="server" PathSeparator=" >
" RenderCurrentNodeAsLink="false"> </asp:SiteMapPath>

<hr />

<asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">

</asp:ContentPlaceHolder>
```

7. **List and explain any 5 templates to create ASP.NET applications.**

- You can create different types of ASP.NET applications by using templates in Visual Studio.

ASP.NET Web Forms Site: This creates a full-featured ASP.NET website, with its basic infrastructure already in place.

- This website includes a master page that defines the overall layout (with a header, footer, and menu bar) and two ready-made web pages, named default.aspx.
- It also includes an accounts folder with pages for registration, login and password changing.
- After you've read this book and learned the details about all ASP.NET web form features, you may choose to use this template as a starting point for a new site.

ASP.NET website (Razor): This creates a website that uses a ASP.NET model view controller (MVC) approach, rather than the web forms model.

- The ASP.NET MVC model offers some benefits that ASP.NET web forms can't match - for example, It give developers more control over the way HTML is generated, and it creates websites that are far easier to use with automated testing tools.
- The disadvantage is that ASP.NET MVC site sacrifice some of traditional ASP.NET convenience and ease of use.

ASP.NET Empty website: This creates a nearly empty website.

- It includes a stripped down web.config configuration file, and nothing else. Of Course, it's easy to fill in the pieces you need as you start coding.
- This is the best starting point to learn about ASP.NET, because you won't have any extra, unnecessary, auto generated files.

ASP.NET Dynamic data entities website: This creates an ASP.NET website that uses the ASP.NET dynamic data feature.

- There are actually two dynamic data templates, which you slightly different approaches to communicating with your database.

ASP.NET 4.5 in C#.WCF Service: This creates a WCF service- a library of server- side methods that remote clients (for example, windows applications) can call.

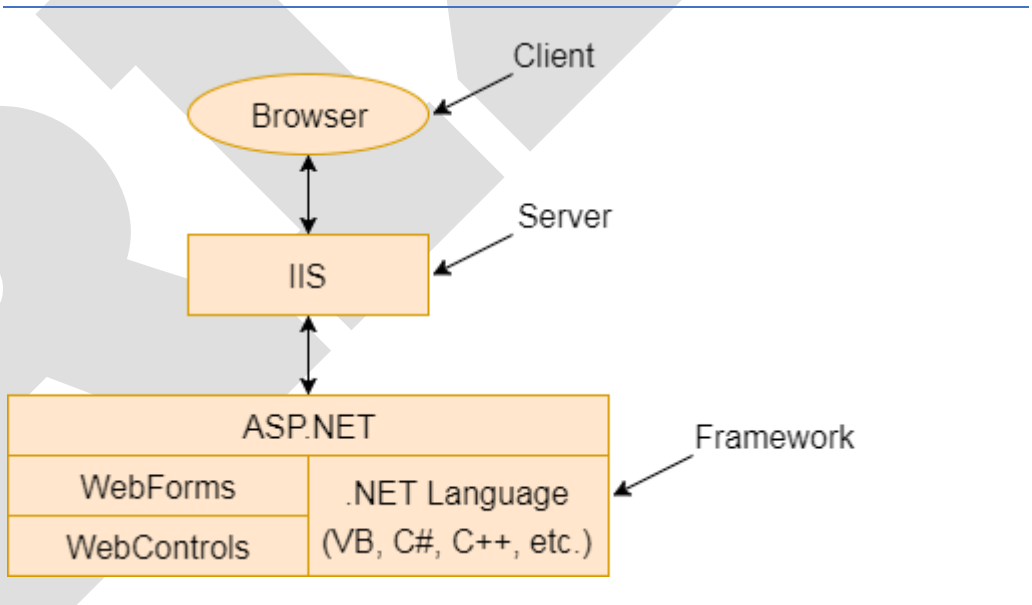
ASP.NET Reports web site: This creates an ASP.NET website that uses the report view control and SQL server reporting services (a tool for generation database reports that can be viewed and managed over the web). The ASP.NET crystal reports website template provides a similar service, but it uses the competing crystals reports software.

8. Explain Anatomy of a Webform.

Web forms are web pages built on the asp.net technology. it executes on the serve and generates output to the browser. it is compatible to any browser to any language supported by .net common language runtime. it is flexible and allows us to create and add custom controls.

We can use visual studio to create asp.net web forms. it is an ide (integrated development web forms are web pages built on the asp.net technology. it executes on the server environment) that allows us to drag and drop server controls to the web forms. it also allows us to set properties, events and methods for the controls. to write business logic, we can choose any .net language like: visual basic or visual c#.

Web forms are made up of two components: the visual portion (the aspx file), and the code behind the form, which resides in a separate class file.



The main purpose of web forms is to overcome the limitations of asp and separate view from the application logic.

ASP.NET provides the awesome features to create and develop web application. here, some of the features of web forms.

- server controls
- master pages
- working with data
- membership
- client script and client frameworks
- routing
- state management
- security
- performance
- error handling

9. Explain any five properties of ListBox and DropDownList controls.

ListBox :

ListBox control is an asp.net web server control. ListBox control used to store the multiple items and allow user to select multiple item from listbox control. ListBox control is same as dropdownlist control.

The dropdownlist control allow user to select maximum only one item at a time, on other hand listbox control allow user to select multiple items same time. so we can also say listbox is a **multi row selection box control**.

Some important Properties of ListBox Control :

PROPERTY	DESCRIPTION
SelectionMode	Single or Multiple. If multiple, it allows user to select multiple items from the list by holding Ctrl or Shift key.
SelectedValue	Get the value of the Selected item from the dropdown box.
SelectedIndex	Gets or Sets the index of the selected item in the dropdown box.
SelectedItem	Gets the selected item from the list.
Items	Gets the collection of items from the dropdown box.
DataTextField	Name of the data source field to supply the text of the items. (No need to set when you are adding items directly into .aspx page.)
DataValueField	Name of the data source field to supply the value of the items. (No need

	to set when you are adding items directly into .aspx page.)
DataSourceID	ID of the datasource component to provide data. (Only used when you have any DataSource component on the page, like SqlDataSource, AccessDataSource etc.)
DataSource	The datasource that populates the items in the listbox box. (Generally used when you are dynamically generating the items from Database.)
AutoPostBack	true or false. If true, the form is automatically posted back to the server when user changes the dropdown list selection. It will also fire OnSelectedIndexChanged method.

Example:

```
<asp:ListBox ID="ListBox3" runat="server" DataTextField="Name"
DataValueField="ID" OnSelectedIndexChanged="GivePostBackResult"
AutoPostBack="True" />
```

DropDownList :

DropDownList control is used to give a single select option to the user from multiple listed items. You can specify its height and width in pixel by setting its height and width but you will not be able give mutliple select option to the user. When it is rendered on the page, it is implemented through <select/> HTML tag. It is also called as Combo box.

Its properties like BackColor, ForeColor etc. are implemented through style properites of . It has less property to decorate in comparison with other controls. There is no property like BorderStyle, BorderWidth. in DropDownList control.

Some important Properties of DropDownList Control :

PROPERTY	DESCRIPTION
SelectedIndex	Gets or Sets the index of the selected item in the dropdown box.
SelectedItem	Gets the selected item from the list.
Items	Gets the collection of items from the dropdown box.
DataTextField	Name of the data source field to supply the text of the items. (No need to set when you are adding items directly into .aspx page.)
DataValueField	Name of the data source field to supply the value of the items. (No need to set when you are adding items directly into .aspx page.)
DataSourceID	ID of the datasource component to provide data. (Only used when you have any DataSource component on the page, like SqlDataSource, AccessDataSource etc.)
DataSource	The datasource that populates the items in the dropdown box. (Generally used when you are dynamically generating the items from Database.)
AutoPostBack	true or false. If true, the form is automatically posted back to the server when user changes the dropdown list selection. It will also fire

	OnSelectedIndexChanged method.

Example:

```
<asp:DropDownList ID="DropDownList1" runat="server" DataTextField="Name"
DataValueField="ID" OnSelectedIndexChanged="GivePostBackResult"
AutoPostBack="True" />
```

10. Brief about Graphics class and it's any 5 methods.

It's essential that programmers understand the graphics class before they attempt to draw images via java. the graphics class provides the framework for all graphics operations within the awt. it plays two different, but related, roles. first, it is the graphics context. the graphics context is information that will affect drawing operations. this includes the background and foreground colors, the font, and the location and dimensions of the clipping rectangle (the region of a component in which graphics can be drawn). it even includes information about the eventual destination of the graphics operations themselves (screen or image). second, the graphics class provides methods for drawing simple geometric shapes, text, and images to the graphics destination. all output to the graphics destination occurs via an invocation of one of these methods.

In order to draw, a program requires a valid graphics context (represented by an instance of the graphics class). because the graphics class is an abstract base class, it cannot be instantiated directly. an instance is typically created by a component, and handed to the program as an argument to a component's

DRAWING BEGINS WITH THE SYSTEM.DRAWING.GRAPHICSCLASS;

METHOD OF GRAPHICS CLASS:

METHODS	DESCRIPTION
Clear	Clears the entire drawing surface and fills it with the specified background color.
FromHwnd	Creates a new Graphics object from a window handle.
FromImage	Creates a new Graphics object from an Image object.
IsVisible	Returns true if a point is within the visible clip region.
MeaureString	Measures a string when drawn with the specified Font object.
SetClip	Sets the clipping region to the Clip property.