Paper - Solution

Q-1 Explain the following Controls in details (properties, methods, events and suitable example)

- **RadioButtonList Control.**

RadioButtonList is the combination of ListBox and RadioButton. Each list item has the separate RadioButton. Multiple RadioButton can be checked in a group at a time.

The RadioButtonList control supports the following properties.

- **AutoPostBack :** Enables you to post the form containing the RadioButtonList back to the server automatically when the RadioButton is checked or unchecked.
- **CellPadding :** Sets the number of pixels between the border and a particular RadioButton.
- **CellSpacing :** Sets the number of pixels between individual RadioButtons in the RadioButtonList.
- **Items :** Represents the collection of items in the RadioButtonList.
- **RepeatColumns :** Determines the number of columns used to display the RadioButtons.
- **RepeatDirection :** Indicates the direction the RadioButtons should be repeated. Possible values are : Horizontal & Verticle
- **RepeatLayout :** Determines how the RadioButtons are formatted.
  Possible values are : Table (Default value) & Flow .
- **Selected Index :** Specifies the index number of the currently checked RadioButton.
- **Selected Item :** Represents the selected RadioButton.
- **Selected Value :** Gets or Sets the selected RadioButton.
- **Text Align :** Indicates the alignment of the RadioButton label relative to the RadioButton . Possible values are : Left & Right .
- **DataSource :** Indicates the datasource for the items in the RadioButtonList.

The RadioButtonList control also supports the following method:

- **DataBind :** Binds the RadioButtonList to its datasource & loads the items from the data source into the ListItem collection.
- **OnSelectedIndexChanged :** Raises SelectedIndexChanged event.

The RadioButtonList control supports the following event:

- **SelectedIndexChanged :** Raised when new RadioButton is checked or unchecked.

Example :

```
<asp:RadioButtonList ID="RadioButtonList1" runat="server"
CellSpacing="30"   RepeatColumns="3" RepeatDirection="”
Verticle”>
        <asp:ListItem>pink</asp:ListItem>
        <asp:ListItem>red</asp:ListItem>
        <asp:ListItem>blue</asp:ListItem>
</asp:RadioButtonList>
```

Or You can add/remove items to the RadioButtonList by codding at runtime like :

```
public partial class _Default : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
      if (Page.IsPostBack == false)
      {
         RadioButtonList 1.Items.Add("Purple");
         RadioButtonList 1.Items.Add("Black");
         RadioButtonList 1.Items.Remove("Pink");
      }
   }
}
```

Or you can also set the datasource property like :

```
using System.Collections;

public partial class _Default : System.Web.UI.Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
      if (Page.IsPostBack == false)
      {
         ArrayList a = new ArrayList();
         a.Add("Red");
         a.Add("Blue");
         a.Add("Green");

         RadioButtonList 1.DataSource=a;
         RadioButtonList 1.DataBind();
      }
   }
```

- **DropDownList Control.**

The DropDownList control is the combination of Listbox and Textbox.

Properties :

AccessKey :

AutoPostBack :

BackColor :

DataSourceID :

Items :

DataTextField : It Specifies the field(column) of datasource which

provides the items for dropdownlist .

TabIndex :

Tooltip :

Enabled :

Visible :

Methods :

- DataBind()
- OnSelectedIndexChanged()

Event :

- SelectedIndexChanged :

Example :

```
<asp:DropDownList ID="DropDownList1" runat="server" >
    <asp:ListItem>AA</asp:ListItem>
    <asp:ListItem>BB</asp:ListItem>
    <asp:ListItem>CC</asp:ListItem>
    <asp:ListItem>DD</asp:ListItem>
    <asp:ListItem>EE</asp:ListItem>
</asp:DropDownList>
```

- **LinkButton Control.**

The LinkButton control, like the Button control, enables you to post a form to the server. Unlike a Button control, however, the LinkButton control renders a link instead of a push button.

Behind the scenes, the LinkButton control uses JavaScript to post the form back to the server. The hyperlink rendered by the LinkButton control looks like this:

Submit

Clicking the LinkButton invokes the JavaScript __doPostBack() method, which posts the form to the server. When the form is posted, the values of all the other form fields in the page are also posted to the server.

The LinkButton control supports the following properties (this is not a complete list):

- **AccessKey :** Enables you to specify a key that navigates to the Button control.
- **BackColor :** Enables you to change the background color of the label.
- **BorderColor :** Enables you to set the color of a border rendered around the label.
- **BorderStyle :** Enables you to display a border around the label. Possible values are NotSet, None, Dotted, Dashed, Solid, Double, Groove, Ridge, Inset, and Outset.
- **BorderWidth :** Enables you to set the size of a border rendered around the label.
- **Font** : Enables you to set the label's font properties.
- **ForeColor :** Enables you to set the color of the content rendered by the label.

- **Enabled :** Enables you to disable the LinkButton control.
- **OnClientClick :** Enables you to specify a client-side script that executes when the LinkButton is clicked.
- **PostBackUrl :** Enables you to post a form to a particular page.
- **TabIndex :** Enables you to specify the tab order of the LinkButton control.
- **Text :** Enables you to label the LinkButton control.

The LinkButton control also supports the following method: