Data Bound Controls in Website

1. Create an employee table with following fields. (Eid, Fangame, M_Name, S_Name, Address, City, gender, salary, Contact, Email-id)

Source code:

```
CREATE TABLE Employee (
    E_id INT PRIMARY KEY,
    F_Name VARCHAR(50),
    M_Name VARCHAR(50),
    S_Name VARCHAR(50),
    Address VARCHAR(100),
    City VARCHAR(50),
    Gender CHAR(1),
    Salary DECIMAL(10, 2),
    Contact VARCHAR(15),
    Email_id VARCHAR(100)
);
```

2. Display employee table data in grid view with edit, update and delete options also allow paging for grid view.

Source code:

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Web.UI.WebControls;

public partial class Default: System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            BindGridView();
        }
    }

    private void BindGridView()
    {
```

```
        string connectionString = "YourConnectionStringHere";

        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            connection.Open();

            string query = "SELECT * FROM Employee";

            SqlDataAdapter adapter = new SqlDataAdapter(query, connection);
            DataTable dataTable = new DataTable();
            adapter.Fill(dataTable);

            GridView1.DataSource = dataTable;
            GridView1.DataBind();
        }
    }

    protected void GridView1_PageIndexChanging(object sender, GridViewPageEventArgs e)
    {
        GridView1.PageIndex = e.NewPageIndex;
        BindGridView();
    }

protected void GridView1_RowEditing(object sender, GridViewEditEventArgs e)
{
    GridView1.EditIndex = e.NewEditIndex;
    BindGridView();
}

protected void GridView1_RowUpdating (object sender, GridViewUpdateEventArgs e)
{
    // Your update logic here

    // Exit edit mode
    GridView1.EditIndex = -1;
    BindGridView();
}

protected void GridView1_RowCancelingEdit(object sender, GridViewCancelEditEventArgs e)
{
    GridView1.EditIndex = -1;
    BindGridView();
}
```

```
protected void GridView1_RowDeleting (object sender, GridViewDeleteEventArgs e)
{
    // Your delete logic here

    BindGridView();
}

}
```

Output:

```
+----+--------+--------+--------+-----------+--------+--------+----------+-----------------+-------------------+
| E_id| F_Name | M_Name | S_Name | Address   | City   | Gender | Salary   | Contact         |
Email-id         |
+----+--------+--------+--------+-----------+--------+--------+----------+-----------------+-------------------+
| 1   | Jaymin   | M   | Valaki | Ahmedabad  | City1  | M  | 50000.00 | 123-456-7890 |
jay1820@example.com |
| 2   | Jane   | M      | Smith  | Address2  | City2  | F      | 60000.00 | 987-654-3210 |
jane@example.com |
| 3   | Bob    | A      | Johnson| Address3  | City3  | M      | 55000.00 | 456-789-1230 |
bob@example.com  |
+----+--------+--------+--------+-----------+--------+--------+----------+-----------------+-------------------+
```

3. Where E_id, F_Name, M_Name, S_Name are read-only fields(Non-editable)

Source code:

Defalt.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Employee List</title>
</head>
<body>
    <form id="form1" runat="server">
        <h1>Employee List</h1>
        <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
AllowPaging="True" PageSize="10"
```

```
        OnPageIndexChanging="GridView1_PageIndexChanging"
OnRowEditing="GridView1_RowEditing" OnRowUpdating="GridView1_RowUpdating"
        OnRowCancelingEdit="GridView1_RowCancelingEdit"
OnRowDeleting="GridView1_RowDeleting">
        <Columns>
            <asp:TemplateField HeaderText="Employee ID">
                <ItemTemplate>
                    <asp:Label ID="lblE_id" runat="server" Text='<%# Eval("E_id")
%>'></asp:Label>
                </ItemTemplate>
            </asp:TemplateField>
            <asp:TemplateField HeaderText="First Name">
                <ItemTemplate>
                    <asp:Label ID="lblF_Name" runat="server" Text='<%# Eval("F_Name")
%>'></asp:Label>
                </ItemTemplate>
            </asp:TemplateField>
            <asp:TemplateField HeaderText="Middle Name">
                <ItemTemplate>
                    <asp:Label ID="lblM_Name" runat="server" Text='<%# Eval("M_Name")
%>'></asp:Label>
                </ItemTemplate>
            </asp:TemplateField>
            <asp:TemplateField HeaderText="Last Name">
                <ItemTemplate>
                    <asp:Label ID="lblS_Name" runat="server" Text='<%# Eval("S_Name")
%>'></asp:Label>
                </ItemTemplate>
            </asp:TemplateField>
            <asp:BoundField DataField="Address" HeaderText="Address" />
            <asp:BoundField DataField="City" HeaderText="City" />
            <asp:BoundField DataField="Gender" HeaderText="Gender" />
            <asp:BoundField DataField="Salary" HeaderText="Salary" />
            <asp:BoundField DataField="Contact" HeaderText="Contact" />
            <asp:BoundField DataField="Email_id" HeaderText="Email ID" />
            <asp:CommandField ShowEditButton="True" ShowDeleteButton="True" />
        </Columns>
    </asp:GridView>
  </form>
</body>
</html>
```

Default.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            BindGridView();
        }
    }

    private void BindGridView()
    {
        // Replace with your connection string
        string connectionString = "YourConnectionStringHere";

        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            connection.Open();

            // SQL query to retrieve data from the Employee table
            string query = "SELECT * FROM Employee";

            SqlDataAdapter adapter = new SqlDataAdapter(query, connection);
            DataTable dataTable = new DataTable();
            adapter.Fill(dataTable);

            GridView1.DataSource = dataTable;
            GridView1.DataBind();
        }
    }

    protected void GridView1_PageIndexChanging(object sender, GridViewPageEventArgs e)
    {
        GridView1.PageIndex = e.NewPageIndex;
        BindGridView();
    }

    protected void GridView1_RowEditing(object sender, GridViewEditEventArgs e)
```

```
    {
        // Cancel editing for the non-editable fields (E_id, F_Name, M_Name, S_Name)
        GridView1.EditIndex = e.NewEditIndex;
        BindGridView();
    }

    protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs e)
    {
        // Implement your update logic here

        GridView1.EditIndex = -1; // Exit edit mode
        BindGridView();
    }

    protected void GridView1_RowCancelingEdit(object sender, GridViewCancelEditEventArgs
e)
    {
        GridView1.EditIndex = -1;
        BindGridView();
    }

    protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
    {
        // Implement your delete logic here

        BindGridView();
    }
}
```

Output:

```
+----+--------+--------+--------+----------+--------+--------+----------+----------------+-----------------+
| E_id| F_Name | M_Name | S_Name | Address   | City   | Gender | Salary   | Contact        |
Email-id         |
+----+--------+--------+--------+----------+--------+--------+----------+----------------+-----------------+
| 1   | John   | M      | Doe    | Address1  | City1  | M      | 50000.00 | 123-456-7890 |
john@example.com |
| 2   | Jane   | M      | Smith  | Address2  | City2  | F      | 60000.00 | 987-654-3210 |
jane@example.com |
| 3   | Bob    | A      | Johnson| Address3  | City3  | M      | 55000.00 | 456-789-1230 |
bob@example.com  |
+----+--------+--------+--------+----------+--------+--------+----------+----------------+-----------------+
```

4. Use table created in program 1. Display employee table data in Details view with add new, update and delete options. Where E_id, F_Name, M_ Name, S_Name are read-only fields(Non-editable).

Source code:

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Employee Details</title>
</head>
<body>
   <form id="form1" runat="server">
      <h1>Employee Details</h1>
      <asp:DetailsView ID="DetailsView1" runat="server" AutoGenerateRows="False"
OnModeChanging="DetailsView1_ModeChanging">
         <Fields>
            <asp:TemplateField HeaderText="Employee ID">
               <ItemTemplate>
                  <asp:Label ID="lblE_id" runat="server" Text='<%# Bind("E_id")
%>'></asp:Label>
               </ItemTemplate>
            </asp:TemplateField>
            <asp:TemplateField HeaderText="First Name">
               <ItemTemplate>
                  <asp:Label ID="lblF_Name" runat="server" Text='<%# Bind("F_Name")
%>'></asp:Label>
               </ItemTemplate>
            </asp:TemplateField>
            <asp:TemplateField HeaderText="Middle Name">
               <ItemTemplate>
                  <asp:Label ID="lblM_Name" runat="server" Text='<%# Bind("M_Name")
%>'></asp:Label>
               </ItemTemplate>
            </asp:TemplateField>
            <asp:TemplateField HeaderText="Last Name">
               <ItemTemplate>
```

```
                <asp:Label ID="lblS_Name" runat="server" Text='<%# Bind("S_Name")
%>'></asp:Label>
            </ItemTemplate>
        </asp:TemplateField>
        <asp:TemplateField HeaderText="Address">
            <EditItemTemplate>
                <asp:TextBox ID="txtAddress" runat="server" Text='<%# Bind("Address")
%>'></asp:TextBox>
            </EditItemTemplate>
            <InsertItemTemplate>
                <asp:TextBox ID="txtNewAddress" runat="server"></asp:TextBox>
            </InsertItemTemplate>
            <ItemTemplate>
                <asp:Label ID="lblAddress" runat="server" Text='<%# Bind("Address")
%>'></asp:Label>
            </ItemTemplate>
        </asp:TemplateField>
        <!-- Add other editable fields here -->
        <asp:TemplateField HeaderText="City">
            <!-- Similar template field for City -->
        </asp:TemplateField>
        <asp:CommandField ShowEditButton="True" ShowDeleteButton="True" />
    </Fields>
</asp:DetailsView>
</form>
</body>
</html>
```

Default.aspx.cs

```
using System;
using System.Data;
using System.Data.SqlClient;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    // Define a connection string to your database
    private string connectionString = "YourConnectionStringHere";

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
```

```
            BindDetailsView();
        }
    }

    private void BindDetailsView()
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            connection.Open();
            string query = "SELECT * FROM Employee WHERE E_id = @E_id"; // Replace with
your condition

            SqlCommand command = new SqlCommand(query, connection);
            command.Parameters.AddWithValue("@E_id", 1); // Example: Assuming you want to
retrieve a specific employee with E_id = 1

            SqlDataReader reader = command.ExecuteReader(CommandBehavior.SingleRow);

            if (reader.Read())
            {
                DetailsView1.DataSource = reader;
                DetailsView1.DataBind();
            }
        }
    }

    protected void DetailsView1_ModeChanging(object sender, DetailsViewModeEventArgs e)
    {
        DetailsView1.ChangeMode(e.NewMode);
        BindDetailsView();
    }

    protected void DetailsView1_ItemInserting(object sender, DetailsViewInsertEventArgs e)
    {
        // Implement the insert operation here
        string firstName = e.Values["F_Name"].ToString();
        string middleName = e.Values["M_Name"].ToString();
        string lastName = e.Values["S_Name"].ToString();
        string address = e.Values["Address"].ToString();
        string city = e.Values["City"].ToString();
        // Get other values as needed

        using (SqlConnection connection = new SqlConnection(connectionString))
        {
```

```
            connection.Open();
            string insertQuery = "INSERT INTO Employee (F_Name, M_Name, S_Name, Address,
City) VALUES (@F_Name, @M_Name, @S_Name, @Address, @City)";

            SqlCommand insertCommand = new SqlCommand(insertQuery, connection);
            insertCommand.Parameters.AddWithValue("@F_Name", firstName);
            insertCommand.Parameters.AddWithValue("@M_Name", middleName);
            insertCommand.Parameters.AddWithValue("@S_Name", lastName);
            insertCommand.Parameters.AddWithValue("@Address", address);
            insertCommand.Parameters.AddWithValue("@City", city);
            // Add parameters for other columns as needed

            insertCommand.ExecuteNonQuery();
        }

        DetailsView1.ChangeMode(DetailsViewMode.ReadOnly);
        BindDetailsView();
    }

    protected void DetailsView1_ItemUpdating(object sender, DetailsViewUpdateEventArgs e)
    {
        // Implement the update operation here
        int employeeId = Convert.ToInt32(e.Keys["E_id"]);
        string firstName = e.NewValues["F_Name"].ToString();
        string middleName = e.NewValues["M_Name"].ToString();
        string lastName = e.NewValues["S_Name"].ToString();
        string address = e.NewValues["Address"].ToString();
        string city = e.NewValues["City"].ToString();
        // Get other values as needed

        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            connection.Open();
            string updateQuery = "UPDATE Employee SET F_Name = @F_Name, M_Name =
@M_Name, S_Name = @S_Name, Address = @Address, City = @City WHERE E_id =
@E_id";

            SqlCommand updateCommand = new SqlCommand(updateQuery, connection);
            updateCommand.Parameters.AddWithValue("@F_Name", firstName);
            updateCommand.Parameters.AddWithValue("@M_Name", middleName);
            updateCommand.Parameters.AddWithValue("@S_Name", lastName);
            updateCommand.Parameters.AddWithValue("@Address", address);
            updateCommand.Parameters.AddWithValue("@City", city);
            updateCommand.Parameters.AddWithValue("@E_id", employeeId);
```

```
            // Add parameters for other columns as needed

            updateCommand.ExecuteNonQuery();
        }

        DetailsView1.ChangeMode(DetailsViewMode.ReadOnly);
        BindDetailsView();
    }

    protected void DetailsView1_ItemDeleting(object sender, DetailsViewDeleteEventArgs e)
    {
        // Implement the delete operation here
        int employeeId = Convert.ToInt32(e.Keys["E_id"]);

        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            connection.Open();
            string deleteQuery = "DELETE FROM Employee WHERE E_id = @E_id";

            SqlCommand deleteCommand = new SqlCommand(deleteQuery, connection);
            deleteCommand.Parameters.AddWithValue("@E_id", employeeId);

            deleteCommand.ExecuteNonQuery();
        }

        DetailsView1.ChangeMode(DetailsViewMode.ReadOnly);
        BindDetailsView();
    }
}
```

Output:

```
+-----+---------+---------+---------+-----------+---------+
| E_id| F_Name | M_Name | S_Name | Address   | City    |
+-----+---------+---------+---------+-----------+---------+
| 1   | John   | M      | Doe    | Address1  | City1  |
| 2   | Jane   | M      | Smith | Address2  | City2   |
| 3   | Bob    | A      | Johnson| Address3  | City3   |
+-----+---------+---------+---------+-----------+---------+
```

5. Using the table created in above programs, demonstrate the use of Data pager. (Data pager works with list view control only.)

Source code:

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeFile="Default.aspx.cs"
Inherits="_Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Employee List</title>
</head>
<body>
    <form id="form1" runat="server">
        <h1>Employee List</h1>

        <!-- ListView to display employee data -->
        <asp:ListView ID="ListView1" runat="server"
OnPagePropertiesChanging="ListView1_PagePropertiesChanging">
            <ItemTemplate>
                <tr>
                    <td><%# Eval("E_id") %></td>
                    <td><%# Eval("F_Name") %></td>
                    <td><%# Eval("M_Name") %></td>
                    <td><%# Eval("S_Name") %></td>
                    <td><%# Eval("Address") %></td>
                    <td><%# Eval("City") %></td>
                </tr>
            </ItemTemplate>
            <LayoutTemplate>
                <table>
                    <thead>
                        <tr>
                            <th>Employee ID</th>
                            <th>First Name</th>
                            <th>Middle Name</th>
                            <th>Last Name</th>
                            <th>Address</th>
                            <th>City</th>
                        </tr>
                    </thead>
                    <tbody>
                        <asp:PlaceHolder runat="server" ID="itemPlaceholder"></asp:PlaceHolder>
                    </tbody>
```

```
            </table>
        </LayoutTemplate>
    </asp:ListView>

    <!-- DataPager for paging through the ListView -->
    <asp:DataPager ID="DataPager1" runat="server" PagedControlID="ListView1"
PageSize="3">
        <Fields>
            <asp:NextPreviousPagerField ButtonType="Button" ShowNextPageButton="true"
ShowPreviousPageButton="true" />
        </Fields>
    </asp:DataPager>
  </form>
</body>
</html>
```

Default.aspx.cs

```
using System;
using System.Web.UI.WebControls;

public partial class _Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            BindListView();
        }
    }

    private void BindListView()
    {
        // Simulated employee data (replace with database retrieval)
        var employeeData = new[]
        {
            new { E_id = 1, F_Name = "John", M_Name = "M", S_Name = "Doe", Address =
"Address1", City = "City1" },
            new { E_id = 2, F_Name = "Jane", M_Name = "M", S_Name = "Smith", Address =
"Address2", City = "City2" },
            new { E_id = 3, F_Name = "Bob", M_Name = "A", S_Name = "Johnson", Address =
"Address3", City = "City3" }
            // Add more data as needed
        };
```

```
        ListView1.DataSource = employeeData;
        ListView1.DataBind();
    }

    protected void ListView1_PagePropertiesChanging (object
sender,PagePropertiesChangingEventArgs e)
    {
        DataPager1.SetPageProperties(e.StartRowIndex, e.MaximumRows, false);
        BindListView();
    }
}
```

Output:

```
+-------------+-----------+----------+-----------+-----------+---------+
| Employee ID| First Name| Middle Name| Last Name| Address   | City    |
+-------------+-----------+----------+-----------+-----------+---------+
| 1          | John      | M        | Doe       | Address1  | City1   |
| 2          | Jane      | M        | Smith     | Address2  | City2   |
| 3          | Bob       | A        | Johnson   | Address3  | City3   |
+-------------+-----------+----------+-----------+-----------+---------+
```