Slip 1

Q.1) Write a web application in asp.net to implement concept of state management using View State and query string. [10 Marks]

Solution:

StateManagement.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="StateManagement.aspx.cs"
Inherits="WebApplication1.StateManagement" %>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>State Management Demo</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h2>State Management Demo</h2>
      <asp:TextBox ID="txtViewState" runat="server"></asp:TextBox>
      <br />
      <asp:Button ID="btnSaveViewState" runat="server" Text="Save ViewState"
OnClick="btnSaveViewState_Click" />
      <br />
      <asp:Button ID="btnRetrieveViewState" runat="server" Text="Retrieve ViewState"
OnClick="btnRetrieveViewState_Click" />
      <br />
      <hr />
      <asp:HyperLink ID="hlQueryString" runat="server"
NavigateUrl="QueryStringPage.aspx">Go to QueryString Page</asp:HyperLink>
    </div>
  </form>
</body>
</html>
```

StateManagement.aspx.cs:

```
using System;
namespace WebApplication1
{
  public partial class StateManagement : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void btnSaveViewState_Click(object sender, EventArgs e)
    {
      ViewState["UserData"] = txtViewState.Text; // Save data to ViewState
      txtViewState.Text = ""; // Clear the textbox after saving
    }
    protected void btnRetrieveViewState_Click(object sender, EventArgs e)
    {
      if (ViewState["UserData"] != null)
      {
        txtViewState.Text = ViewState["UserData"].ToString(); // Retrieve data from ViewState
```

```
            }
        }
    }
}
```

QueryStringPage.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="QueryStringPage.aspx.cs"
Inherits="WebApplication1.QueryStringPage" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Query String Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Query String Page</h2>
            <asp:Label ID="lblQueryStringData" runat="server"></asp:Label>
        </div>
    </form>
</body>
</html>
```

QueryStringPage.aspx.cs:

```
using System;

namespace WebApplication1
{
    public partial class QueryStringPage : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                if (!string.IsNullOrEmpty(Request.QueryString["data"]))
                {
                    string data = Request.QueryString["data"]; // Retrieve data from query string
                    lblQueryStringData.Text = "Data from QueryString: " + data;
                }
            }
        }
    }
}
```

Q.2) Create an ASP.Net application which displays 5-10 quiz questions, each with four possible options and exactly one correct option; the application then counts and presents the user with

the number of correct answers, and shows the result to SPPU M.Sc. Computer Science Syllabus 2023-24 of 5 user’s.

Solution:

Quiz.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Quiz.aspx.cs"
Inherits="QuizApplication.Quiz" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Quiz</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h1>Quiz</h1>
      <asp:Label ID="lblQuestion" runat="server"></asp:Label>
      <br />
      <asp:RadioButtonList ID="rbOptions" runat="server" RepeatDirection="Vertical">
      </asp:RadioButtonList>
      <br />
      <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />
      <br />
      <asp:Label ID="lblResult" runat="server" Visible="false"></asp:Label>
    </div>
  </form>
</body>
</html>
```

Quiz.aspx.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web.UI.WebControls;

namespace QuizApplication
{
  public partial class Quiz : System.Web.UI.Page
  {
    private Dictionary<string, string> quizData = new Dictionary<string, string>()
    {
      { "What is the capital of France?", "Paris" },
      { "Which planet is known as the Red Planet?", "Mars" },
      // Add more questions and correct answers here
    };

    protected void Page_Load(object sender, EventArgs e)
    {
      if (!IsPostBack)
      {
          LoadQuestion();
      }
    }
    protected void LoadQuestion()
    {
      Random random = new Random();
      var question = quizData.ElementAt(random.Next(0, quizData.Count));
      lblQuestion.Text = question.Key;
      rbOptions.Items.Clear();
      rbOptions.Items.Add(new ListItem(question.Value));
      var otherOptions = quizData.Values.Except(new[] { question.Value }).OrderBy(x =>
random.Next()).Take(3);
      foreach (var option in otherOptions)
      {
          rbOptions.Items.Add(new ListItem(option));
      }
    }
    protected void btnSubmit_Click(object sender, EventArgs e)
    {
      string selectedOption = rbOptions.SelectedItem?.Text;
      string correctAnswer = quizData[lblQuestion.Text];
      if (selectedOption == correctAnswer)
      {
          lblResult.Text = "Correct!";
      }
      else
      {
          lblResult.Text = "Incorrect!";
      }
      lblResult.Visible = true;
    }
  }
}
```

Slip 2

Q.1) Write a web application in ASP.Net take two buttons on the page, a text box to enter string and a label display the text stored from last session.

Solution:

Default.aspx

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="WebApplication1.Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Session Demo</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:TextBox ID="txtInput" runat="server"></asp:TextBox>
            <br />
            <asp:Button ID="btnSave" runat="server" Text="Save" OnClick="btnSave_Click" />
            <asp:Button ID="btnClear" runat="server" Text="Clear" OnClick="btnClear_Click" />
            <br />
            <asp:Label ID="lblStoredText" runat="server"></asp:Label>
        </div>
    </form>
</body>
</html>
```

Default.aspx.cs

```
using System;

namespace WebApplication1
{
    public partial class Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                // Check if text is stored in session
                if (Session["StoredText"] != null)
                {
                    lblStoredText.Text = Session["StoredText"].ToString();
                }
            }
        }

        protected void btnSave_Click(object sender, EventArgs e)
        {
            // Save text to session
            Session["StoredText"] = txtInput.Text;
            lblStoredText.Text = txtInput.Text;
        }

        protected void btnClear_Click(object sender, EventArgs e)
        {
            // Clear text from session
            Session["StoredText"] = null;
            lblStoredText.Text = string.Empty;
            txtInput.Text = string.Empty;
        }
    }
}
```

Q.2) Write an ASP.Net program to display employee details (EmpID, Name, Designation, Joining Date, Mob.no, Gender) from a database and allow editing and deleting information from a GridView.

Solution:

EmployeeDetails.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="EmployeeDetails.aspx.cs"
Inherits="WebApplication1.EmployeeDetails" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Employee Details</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
OnRowEditing="GridView1_RowEditing" OnRowDeleting="GridView1_RowDeleting">
                <Columns>
                    <asp:BoundField DataField="EmpID" HeaderText="EmpID" />
                    <asp:BoundField DataField="Name" HeaderText="Name" />
                    <asp:BoundField DataField="Designation" HeaderText="Designation" />
                    <asp:BoundField DataField="JoiningDate" HeaderText="Joining Date" />
                    <asp:BoundField DataField="MobileNumber" HeaderText="Mobile Number" />
                    <asp:BoundField DataField="Gender" HeaderText="Gender" />
                    <asp:CommandField ShowEditButton="True" ShowDeleteButton="True" />
                </Columns>
            </asp:GridView>
        </div>
    </form>
</body>
</html>
```

EmployeeDetails.aspx.cs:

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

namespace WebApplication1
```

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

        protected void BindGrid()
        {
            string constr =
ConfigurationManager.ConnectionStrings["YourConnectionString"].ConnectionString;
            using (SqlConnection con = new SqlConnection(constr))
            {
                using (SqlCommand cmd = new SqlCommand("SELECT * FROM Employee", con))
                {
                    using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
                    {
                        DataTable dt = new DataTable();
                        sda.Fill(dt);
                        GridView1.DataSource = dt;
                        GridView1.DataBind();
                    }
                }
            }
        }

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

        protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
        {
            int empID = Convert.ToInt32(GridView1.DataKeys[e.RowIndex].Values[0]);
            string constr =
ConfigurationManager.ConnectionStrings["YourConnectionString"].ConnectionString;
            using (SqlConnection con = new SqlConnection(constr))
            {
                using (SqlCommand cmd = new SqlCommand("DELETE FROM Employee WHERE
EmpID = @EmpID", con))
                {
                    cmd.Parameters.AddWithValue("@EmpID", empID);
                    con.Open();
                    cmd.ExecuteNonQuery();
                    con.Close();
                }
            }
            BindGrid();
        }
    }
}
```

Slip 3

Q.1)Write an ASP.net program to transfer items from one ListView to another ListView using ListView.

<u>Solution:</u>

ListViewTransfer.aspx (HTML):

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="ListViewTransfer.aspx.cs"
Inherits="WebApplication1.ListViewTransfer" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>ListView Transfer</title>
    <script>
        function transferItems(sourceID, destID) {
            var source = document.getElementById(sourceID);
            var destination = document.getElementById(destID);

            var sourceItems = source.getElementsByTagName('input');
            for (var i = 0; i < sourceItems.length; i++) {
                var checkbox = sourceItems[i];
                if (checkbox.checked) {
                    var item = checkbox.parentElement.parentElement.cloneNode(true);
                    destination.appendChild(item);
                    checkbox.checked = false;
                }
            }
        }
    </script>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Source ListView</h2>
            <asp:ListView ID="lvSource" runat="server">
                <ItemTemplate>
                    <tr>
                        <td><%# Eval("ItemID") %></td>
                        <td><%# Eval("ItemName") %></td>
                        <td><input type="checkbox" /></td>
                    </tr>
                </ItemTemplate>
            </asp:ListView>
            <br />
            <button type="button" onclick="transferItems('lvSource', 'lvDestination')">Transfer Selected</button>
        </div>
        <div>
            <h2>Destination ListView</h2>
            <asp:ListView ID="lvDestination" runat="server">
                <ItemTemplate>
                    <tr>
                        <td><%# Eval("ItemID") %></td>
                        <td><%# Eval("ItemName") %></td>
                    </tr>
                </ItemTemplate>
            </asp:ListView>
        </div>
    </form>
</body>
</html>
```

ListViewTransfer.aspx.cs (Code-behind):

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

namespace WebApplication1
{
    public partial class ListViewTransfer : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                // Bind source ListView with sample data
                BindSourceListView();
            }
        }

        protected void BindSourceListView()
        {
            // Sample data for source ListView
            List<Item> items = new List<Item>
            {
                new Item { ItemID = 1, ItemName = "Item 1" },
                new Item { ItemID = 2, ItemName = "Item 2" },
                new Item { ItemID = 3, ItemName = "Item 3" },
                new Item { ItemID = 4, ItemName = "Item 4" }
            };

            lvSource.DataSource = items;
            lvSource.DataBind();
        }
    }
    public class Item
    {
        public int ItemID { get; set; }
        public string ItemName { get; set; }
    }
}
```

Q.2) Write an ASP.Net program to display student details (roll number, name, mobile number, address, gender) from a database to a GridView, allowing editing and deleting information directly in the GridView.

Solution:

StudentDetails.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="StudentDetails.aspx.cs"
Inherits="WebApplication1.StudentDetails" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Student Details</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
OnRowEditing="GridView1_RowEditing" OnRowCancelingEdit="GridView1_RowCancelingEdit"
OnRowUpdating="GridView1_RowUpdating" OnRowDeleting="GridView1_RowDeleting">
                <Columns>
                    <asp:BoundField DataField="RollNumber" HeaderText="Roll Number"
ReadOnly="true" />
                    <asp:BoundField DataField="Name" HeaderText="Name" />
                    <asp:BoundField DataField="MobileNumber" HeaderText="Mobile Number" />
                    <asp:BoundField DataField="Address" HeaderText="Address" />
                    <asp:BoundField DataField="Gender" HeaderText="Gender" />
                    <asp:CommandField ShowEditButton="True" ShowCancelButton="True"
ShowDeleteButton="True" />
                </Columns>
            </asp:GridView>
        </div>
    </form>
</body>
</html>
```

StudentDetails.aspx.cs:

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

namespace WebApplication1
```

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

    protected void BindGrid()
    {
      string constr =
ConfigurationManager.ConnectionStrings["YourConnectionString"].ConnectionString;
      using (SqlConnection con = new SqlConnection(constr))
      {
        using (SqlCommand cmd = new SqlCommand("SELECT * FROM Students", con))
        {
          using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
          {
            DataTable dt = new DataTable();
            sda.Fill(dt);
            GridView1.DataSource = dt;
            GridView1.DataBind();
          }
        }
      }
    }

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

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

    protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs e)
    {
      GridViewRow row = GridView1.Rows[e.RowIndex];
      int rollNumber = Convert.ToInt32(GridView1.DataKeys[e.RowIndex].Values[0]);
      string name = (row.FindControl("txtName") as TextBox).Text;
      string mobileNumber = (row.FindControl("txtMobileNumber") as TextBox).Text;
      string address = (row.FindControl("txtAddress") as TextBox).Text;
      string gender = (row.FindControl("txtGender") as TextBox).Text;
```

```
            string constr =
ConfigurationManager.ConnectionStrings["YourConnectionString"].ConnectionString;
            using (SqlConnection con = new SqlConnection(constr))
            {
                using (SqlCommand cmd = new SqlCommand("UPDATE Students SET Name = @Name,
MobileNumber = @MobileNumber, Address = @Address, Gender = @Gender WHERE RollNumber
= @RollNumber", con))
                {
                    cmd.Parameters.AddWithValue("@Name", name);
                    cmd.Parameters.AddWithValue("@MobileNumber", mobileNumber);
                    cmd.Parameters.AddWithValue("@Address", address);
                    cmd.Parameters.AddWithValue("@Gender", gender);
                    cmd.Parameters.AddWithValue("@RollNumber", rollNumber);
                    con.Open();
                    cmd.ExecuteNonQuery();
                    con.Close();
                }
            }
            GridView1.EditIndex = -1;
            BindGrid();
        }

        protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
        {
            int rollNumber = Convert.ToInt32(GridView1.DataKeys[e.RowIndex].Values[0]);
            string constr =
ConfigurationManager.ConnectionStrings["YourConnectionString"].ConnectionString;
            using (SqlConnection con = new SqlConnection(constr))
            {
                using (SqlCommand cmd = new SqlCommand("DELETE FROM Students WHERE
RollNumber = @RollNumber", con))
                {
                    cmd.Parameters.AddWithValue("@RollNumber", rollNumber);
                    con.Open();
                    cmd.ExecuteNonQuery();
                    con.Close();
                }
            }
            BindGrid();
        }
    }
}
```

Slip 4

Q.1) Write an ASP.Net program using validation control to Validate student details form.

Solution:

StudentDetailsForm.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="StudentDetailsForm.aspx.cs"
Inherits="WebApplication1.StudentDetailsForm" %>
```

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Student Details Form</title>
</head>
<body>
   <form id="form1" runat="server">
       <div>
          <h2>Student Details Form</h2>
          <table>
             <tr>
                <td>Roll Number:</td>
                <td><asp:TextBox ID="txtRollNumber" runat="server"></asp:TextBox></td>
                <td><asp:RequiredFieldValidator ID="rfvRollNumber" runat="server"
ControlToValidate="txtRollNumber" ErrorMessage="Roll Number is
required">*</asp:RequiredFieldValidator></td>
             </tr>
             <tr>
                <td>Name:</td>
                <td><asp:TextBox ID="txtName" runat="server"></asp:TextBox></td>
                <td><asp:RequiredFieldValidator ID="rfvName" runat="server"
ControlToValidate="txtName" ErrorMessage="Name is
required">*</asp:RequiredFieldValidator></td>
             </tr>
             <tr>
                <td>Mobile Number:</td>
                <td><asp:TextBox ID="txtMobileNumber" runat="server"></asp:TextBox></td>
                <td><asp:RegularExpressionValidator ID="revMobileNumber" runat="server"
ControlToValidate="txtMobileNumber" ErrorMessage="Invalid Mobile Number"
ValidationExpression="^\d{10}$">*</asp:RegularExpressionValidator></td>
             </tr>
             <tr>
                <td>Address:</td>
                <td><asp:TextBox ID="txtAddress" runat="server"></asp:TextBox></td>
             </tr>
             <tr>
                <td>Gender:</td>
                <td><asp:DropDownList ID="ddlGender" runat="server">
                      <asp:ListItem Text="Select Gender" Value=""></asp:ListItem>
                      <asp:ListItem Text="Male" Value="Male"></asp:ListItem>
                      <asp:ListItem Text="Female" Value="Female"></asp:ListItem>
                   </asp:DropDownList></td>
                <td><asp:RequiredFieldValidator ID="rfvGender" runat="server"
ControlToValidate="ddlGender" ErrorMessage="Gender is
required">*</asp:RequiredFieldValidator></td>
             </tr>
             <tr>
                <td colspan="3">
                   <asp:Button ID="btnSubmit" runat="server" Text="Submit"
OnClick="btnSubmit_Click" />
```

```
                </td>
            </tr>
        </table>
    </div>
  </form>
</body>
</html>
```

StudentDetailsForm.aspx.cs:

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

namespace WebApplication1
{
  public partial class StudentDetailsForm : Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void btnSubmit_Click(object sender, EventArgs e)
    {
      if (Page.IsValid)
      {
        // Validation passed, process form submission
        // For simplicity, you can just display a success message
        Response.Write("Form submitted successfully!");
      }
      else
      {
        // Validation failed, display error messages
        // The error messages will be displayed automatically by validation controls
      }
    }
  }
}
```

Q.2) Create a web application that accepts user ID, name, password, age, and email ID. All Information entry is compulsory. Password should be reconfirmed. Age should be within 21 to 30. Email ID should be valid. User ID should have at least one capital letter and digit, With a length between 7 to 20 characters.

Solution:

RegistrationForm.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="RegistrationForm.aspx.cs"
Inherits="WebApplication1.RegistrationForm" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>User Registration</title>
```

```
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>User Registration</h2>
            <table>
                <tr>
                    <td>User ID:</td>
                    <td><asp:TextBox ID="txtUserID" runat="server"></asp:TextBox></td>
                    <td><asp:RequiredFieldValidator ID="rfvUserID" runat="server"
ControlToValidate="txtUserID" ErrorMessage="User ID is
required">*</asp:RequiredFieldValidator></td>
                    <td><asp:RegularExpressionValidator ID="revUserID" runat="server"
ControlToValidate="txtUserID" ErrorMessage="User ID should have at least one capital letter and
digit, with a length between 7 to 20 characters" ValidationExpression="^(?=.*[A-
Z])(?=.*\d).{7,20}$">*</asp:RegularExpressionValidator></td>
                </tr>
                <tr>
                    <td>Name:</td>
                    <td><asp:TextBox ID="txtName" runat="server"></asp:TextBox></td>
                    <td><asp:RequiredFieldValidator ID="rfvName" runat="server"
ControlToValidate="txtName" ErrorMessage="Name is
required">*</asp:RequiredFieldValidator></td>
                </tr>
                <tr>
                    <td>Password:</td>
                    <td><asp:TextBox ID="txtPassword" runat="server"
TextMode="Password"></asp:TextBox></td>
                    <td><asp:RequiredFieldValidator ID="rfvPassword" runat="server"
ControlToValidate="txtPassword" ErrorMessage="Password is
required">*</asp:RequiredFieldValidator></td>
                </tr>
                <tr>
                    <td>Confirm Password:</td>
                    <td><asp:TextBox ID="txtConfirmPassword" runat="server"
TextMode="Password"></asp:TextBox></td>
                    <td><asp:CompareValidator ID="cvConfirmPassword" runat="server"
ControlToValidate="txtConfirmPassword" ControlToCompare="txtPassword"
ErrorMessage="Passwords do not match">*</asp:CompareValidator></td>
                </tr>
                <tr>
                    <td>Age:</td>
                    <td><asp:TextBox ID="txtAge" runat="server"></asp:TextBox></td>
                    <td><asp:RangeValidator ID="rvAge" runat="server" ControlToValidate="txtAge"
ErrorMessage="Age should be between 21 and 30" MinimumValue="21"
MaximumValue="30">*</asp:RangeValidator></td>
                </tr>
                <tr>
                    <td>Email ID:</td>
                    <td><asp:TextBox ID="txtEmail" runat="server"></asp:TextBox></td>
```

```
          <td><asp:RequiredFieldValidator ID="rfvEmail" runat="server"
ControlToValidate="txtEmail" ErrorMessage="Email ID is
required">*</asp:RequiredFieldValidator></td>
          <td><asp:RegularExpressionValidator ID="revEmail" runat="server"
ControlToValidate="txtEmail" ErrorMessage="Invalid Email ID" ValidationExpression="\w+([-
+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*">*</asp:RegularExpressionValidator></td>
        </tr>
        <tr>
          <td colspan="3">
            <asp:Button ID="btnSubmit" runat="server" Text="Submit"
OnClick="btnSubmit_Click" />
          </td>
        </tr>
      </table>
    </div>
  </form>
</body>
</html>
```

RegistrationForm.aspx.cs:

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

namespace WebApplication1
{
  public partial class RegistrationForm : Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void btnSubmit_Click(object sender, EventArgs e)
    {
      if (Page.IsValid)
      {
        // Validation passed, process form submission
        // For simplicity, you can just display a success message
        Response.Write("Form submitted successfully!");
      }
      else
      {
        // Validation failed, display error messages
        // The error messages will be displayed automatically by validation controls
      }
    }
  }
}
```

Slip 5

Q.1) Create a web application that prompts the user to input their name and a message. Then, concatenate the two items and display them in a label. Use radio buttons and check boxes for format selection, allowing the user to make the label text bold, underlined, or italic, and change its color. Additionally, add a button to display the message in the label, another button to clear the text boxes and label, and third button for an exit option

Solution:

WebForm1.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication1.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Message Formatter</title>
    <style>
        .bold { font-weight: bold; }
        .italic { font-style: italic; }
        .underlined { text-decoration: underline; }
    </style>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Message Formatter</h2>
            <table>
                <tr>
                    <td>Name:</td>
                    <td><asp:TextBox ID="txtName" runat="server"></asp:TextBox></td>
                </tr>
                <tr>
                    <td>Message:</td>
                    <td><asp:TextBox ID="txtMessage" runat="server"></asp:TextBox></td>
                </tr>
                <tr>
                    <td colspan="2">
                        <input type="checkbox" id="chkBold" runat="server" /> Bold
                        <input type="checkbox" id="chkItalic" runat="server" /> Italic
                        <input type="checkbox" id="chkUnderline" runat="server" /> Underline
                    </td>
                </tr>
                <tr>
                    <td>Color:</td>
                    <td>
                        <input type="radio" id="radRed" runat="server" name="colorGroup" value="Red" />
Red
                        <input type="radio" id="radGreen" runat="server" name="colorGroup" value="Green"
/> Green
                        <input type="radio" id="radBlue" runat="server" name="colorGroup" value="Blue" />
Blue
                    </td>
                </tr>
                <tr>
                    <td colspan="2">
                        <asp:Button ID="btnDisplay" runat="server" Text="Display"
OnClick="btnDisplay_Click" />
                        <asp:Button ID="btnClear" runat="server" Text="Clear" OnClick="btnClear_Click" />
                        <asp:Button ID="btnExit" runat="server" Text="Exit" OnClick="btnExit_Click" />
                    </td>
                </tr>
            </table>
            <br />
            <asp:Label ID="lblDisplay" runat="server"></asp:Label>
        </div>
    </form>
</body>
</html>
```

WebForm1.aspx.cs:

```
using System;
using System.Drawing;

namespace WebApplication1
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    protected void btnDisplay_Click(object sender, EventArgs e)
    {
        string name = txtName.Text.Trim();
        string message = txtMessage.Text.Trim();
        string formattedMessage = message;

        // Check formatting options
        if (chkBold.Checked)
            formattedMessage = "<b>" + formattedMessage + "</b>";
        if (chkItalic.Checked)
            formattedMessage = "<i>" + formattedMessage + "</i>";
        if (chkUnderline.Checked)
            formattedMessage = "<u>" + formattedMessage + "</u>";

        // Apply color
        Color textColor = GetSelectedColor();
        string colorCode = ColorTranslator.ToHtml(textColor);
        formattedMessage = $"<font color='{colorCode}'>{formattedMessage}</font>";

        // Display formatted message
        lblDisplay.Text = $"{name}: {formattedMessage}";
    }

    protected void btnClear_Click(object sender, EventArgs e)
    {
        // Clear text boxes and label
        txtName.Text = "";
        txtMessage.Text = "";
        lblDisplay.Text = "";
    }

    protected void btnExit_Click(object sender, EventArgs e)
    {
        // Close the web page
        Response.Redirect("about:blank");
    }

    private Color GetSelectedColor()
    {
        // Get selected color from radio buttons
        if (radRed.Checked)
            return Color.Red;
        else if (radGreen.Checked)
            return Color.Green;
        else if (radBlue.Checked)
            return Color.Blue;
        else
            return Color.Black; // Default color
    }
}
```

```
}
```

Q.2) Write an ASP.Net program to Display Employee details (EmpID, Name, Designation, Joining Date, Mob.no, Gender) from database to gridview and allow Edit, Delete information from GridView.

<u>Solution:</u>

EmployeeDetails.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="EmployeeDetails.aspx.cs"
Inherits="WebApplication1.EmployeeDetails" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Employee Details</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Employee Details</h2>
            <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
OnRowEditing="GridView1_RowEditing" OnRowCancelingEdit="GridView1_RowCancelingEdit"
OnRowUpdating="GridView1_RowUpdating" OnRowDeleting="GridView1_RowDeleting">
                <Columns>
                    <asp:BoundField DataField="EmpID" HeaderText="EmpID" ReadOnly="true" />
                    <asp:BoundField DataField="Name" HeaderText="Name" />
                    <asp:BoundField DataField="Designation" HeaderText="Designation" />
                    <asp:BoundField DataField="JoiningDate" HeaderText="Joining Date" />
                    <asp:BoundField DataField="MobileNumber" HeaderText="Mobile Number" />
                    <asp:BoundField DataField="Gender" HeaderText="Gender" />
                    <asp:CommandField ShowEditButton="True" ShowCancelButton="True"
ShowDeleteButton="True" />
                </Columns>
            </asp:GridView>
        </div>
    </form>
</body>
</html>
```

EmployeeDetails.aspx.cs:

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

namespace WebApplication1
{
    public partial class EmployeeDetails : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
    {
      if (!IsPostBack)
      {
        BindGrid();
      }
    }

    protected void BindGrid()
    {
      string constr =
ConfigurationManager.ConnectionStrings["YourConnectionString"].ConnectionString;
      using (SqlConnection con = new SqlConnection(constr))
      {
        using (SqlCommand cmd = new SqlCommand("SELECT * FROM Employees", con))
        {
          using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
          {
            DataTable dt = new DataTable();
            sda.Fill(dt);
            GridView1.DataSource = dt;
            GridView1.DataBind();
          }
        }
      }
    }

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

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

    protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs e)
    {
      GridViewRow row = GridView1.Rows[e.RowIndex];
      int empID = Convert.ToInt32(GridView1.DataKeys[e.RowIndex].Values[0]);
      string name = (row.FindControl("txtName") as TextBox).Text;
      string designation = (row.FindControl("txtDesignation") as TextBox).Text;
      string joiningDate = (row.FindControl("txtJoiningDate") as TextBox).Text;
      string mobileNumber = (row.FindControl("txtMobileNumber") as TextBox).Text;
      string gender = (row.FindControl("txtGender") as TextBox).Text;

      string constr =
ConfigurationManager.ConnectionStrings["YourConnectionString"].ConnectionString;
      using (SqlConnection con = new SqlConnection(constr))
```

```
            {
                using (SqlCommand cmd = new SqlCommand("UPDATE Employees SET Name =
@Name, Designation = @Designation, JoiningDate = @JoiningDate, MobileNumber =
@MobileNumber, Gender = @Gender WHERE EmpID = @EmpID", con))
                {
                    cmd.Parameters.AddWithValue("@Name", name);
                    cmd.Parameters.AddWithValue("@Designation", designation);
                    cmd.Parameters.AddWithValue("@JoiningDate", joiningDate);
                    cmd.Parameters.AddWithValue("@MobileNumber", mobileNumber);
                    cmd.Parameters.AddWithValue("@Gender", gender);
                    cmd.Parameters.AddWithValue("@EmpID", empID);
                    con.Open();
                    cmd.ExecuteNonQuery();
                    con.Close();
                }
            }
            GridView1.EditIndex = -1;
            BindGrid();
        }

        protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
        {
            int empID = Convert.ToInt32(GridView1.DataKeys[e.RowIndex].Values[0]);
            string constr =
ConfigurationManager.ConnectionStrings["YourConnectionString"].ConnectionString;
            using (SqlConnection con = new SqlConnection(constr))
            {
                using (SqlCommand cmd = new SqlCommand("DELETE FROM Employees WHERE
EmpID = @EmpID", con))
                {
                    cmd.Parameters.AddWithValue("@EmpID", empID);
                    con.Open();
                    cmd.ExecuteNonQuery();
                    con.Close();
                }
            }
            BindGrid();
        }
    }
}
```

Slip 6

Q.1) Write an ASP.NET application to add selected or all records from a listbox. Assume that a list of employees is available in the listbox, with the multi-line property of the textbox set to true.

Solution:

AddEmployees.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="AddEmployees.aspx.cs"
Inherits="WebApplication1.AddEmployees" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Add Employees</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <h2>Add Employees</h2>
         <asp:ListBox ID="ListBoxEmployees" runat="server" SelectionMode="Multiple">
            <asp:ListItem Text="John" Value="1"></asp:ListItem>
            <asp:ListItem Text="Jane" Value="2"></asp:ListItem>
            <asp:ListItem Text="Alice" Value="3"></asp:ListItem>
            <asp:ListItem Text="Bob" Value="4"></asp:ListItem>
            <asp:ListItem Text="Charlie" Value="5"></asp:ListItem>
         </asp:ListBox>
         <br />
         <asp:Button ID="btnAddSelected" runat="server" Text="Add Selected"
OnClick="btnAddSelected_Click" />
         <asp:Button ID="btnAddAll" runat="server" Text="Add All" OnClick="btnAddAll_Click" />
         <br />
         <asp:TextBox ID="txtSelectedEmployees" runat="server" TextMode="MultiLine" Rows="5"
Columns="50"></asp:TextBox>
      </div>
   </form>
</body>
</html>
```

AddEmployees.aspx.cs:

```
using System;
using System.Text;
```

```
namespace WebApplication1
{
    public partial class AddEmployees : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void btnAddSelected_Click(object sender, EventArgs e)
        {
            AddSelectedEmployees();
        }

        protected void btnAddAll_Click(object sender, EventArgs e)
        {
            AddAllEmployees();
        }

        private void AddSelectedEmployees()
        {
            StringBuilder selectedEmployees = new StringBuilder();

            foreach (ListItem item in ListBoxEmployees.Items)
            {
                if (item.Selected)
                {
                    selectedEmployees.AppendLine(item.Text);
                }
            }
            txtSelectedEmployees.Text += selectedEmployees.ToString();
        }
        private void AddAllEmployees()
        {
            StringBuilder allEmployees = new StringBuilder();

            foreach (ListItem item in ListBoxEmployees.Items)
            {
                allEmployees.AppendLine(item.Text);
            }

            txtSelectedEmployees.Text += allEmployees.ToString();
        }
    }
}
```

Q.2) Write an ASP.net program, the user can enter teacher employee information in database and display into the gridview.

Solution:

TeacherInfo.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="TeacherInfo.aspx.cs"
Inherits="WebApplication1.TeacherInfo" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Teacher Employee Information</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h2>Enter Teacher Employee Information</h2>
      <div>
        <label for="txtName">Name:</label>
        <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
      </div>
      <div>
        <label for="txtDesignation">Designation:</label>
        <asp:TextBox ID="txtDesignation" runat="server"></asp:TextBox>
      </div>
      <div>
        <label for="txtJoiningDate">Joining Date:</label>
        <asp:TextBox ID="txtJoiningDate" runat="server"></asp:TextBox>
      </div>
      <div>
        <label for="txtMobileNumber">Mobile Number:</label>
        <asp:TextBox ID="txtMobileNumber" runat="server"></asp:TextBox>
      </div>
      <div>
        <label for="ddlGender">Gender:</label>
        <asp:DropDownList ID="ddlGender" runat="server">
          <asp:ListItem Text="Male" Value="Male"></asp:ListItem>
          <asp:ListItem Text="Female" Value="Female"></asp:ListItem>
        </asp:DropDownList>
      </div>
      <div>
        <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click"
/>
```

```
            </div>
            <hr />
            <h2>Teacher Employee Information</h2>
            <asp:GridView ID="GridViewTeacherInfo" runat="server" AutoGenerateColumns="False">
                <Columns>
                    <asp:BoundField DataField="Name" HeaderText="Name" />
                    <asp:BoundField DataField="Designation" HeaderText="Designation" />
                    <asp:BoundField DataField="JoiningDate" HeaderText="Joining Date" />
                    <asp:BoundField DataField="MobileNumber" HeaderText="Mobile Number" />
                    <asp:BoundField DataField="Gender" HeaderText="Gender" />
                </Columns>
            </asp:GridView>
        </div>
    </form>
</body>
</html>
```

TeacherInfo.aspx.cs:

```
using System;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;

namespace WebApplication1
{
    public partial class TeacherInfo : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                BindGridView();
            }
        }

        protected void btnSubmit_Click(object sender, EventArgs e)
        {
            AddTeacherInfo();
            BindGridView();
            ClearInputs();
        }

        private void BindGridView()
        {
            string constr =
ConfigurationManager.ConnectionStrings["YourConnectionString"].ConnectionString;
            using (SqlConnection con = new SqlConnection(constr))
            {
                using (SqlCommand cmd = new SqlCommand("SELECT * FROM TeacherInfo", con))
                {
                    using (SqlDataAdapter sda = new SqlDataAdapter(cmd))
                    {
```

```
                    DataTable dt = new DataTable();
                    sda.Fill(dt);
                    GridViewTeacherInfo.DataSource = dt;
                    GridViewTeacherInfo.DataBind();
                }
            }
        }
    }

    private void AddTeacherInfo()
    {
        string name = txtName.Text.Trim();
        string designation = txtDesignation.Text.Trim();
        string joiningDate = txtJoiningDate.Text.Trim();
        string mobileNumber = txtMobileNumber.Text.Trim();
        string gender = ddlGender.SelectedValue;

        string constr =
ConfigurationManager.ConnectionStrings["YourConnectionString"].ConnectionString;
        using (SqlConnection con = new SqlConnection(constr))
        {
            using (SqlCommand cmd = new SqlCommand("INSERT INTO TeacherInfo (Name,
Designation, JoiningDate, MobileNumber, Gender) VALUES (@Name, @Designation,
@JoiningDate, @MobileNumber, @Gender)", con))
            {
                cmd.Parameters.AddWithValue("@Name", name);
                cmd.Parameters.AddWithValue("@Designation", designation);
                cmd.Parameters.AddWithValue("@JoiningDate", joiningDate);
                cmd.Parameters.AddWithValue("@MobileNumber", mobileNumber);
                cmd.Parameters.AddWithValue("@Gender", gender);
                con.Open();
                cmd.ExecuteNonQuery();
                con.Close();
            }
        }
    }

    private void ClearInputs()
    {
        txtName.Text = "";
        txtDesignation.Text = "";
        txtJoiningDate.Text = "";
        txtMobileNumber.Text = "";
        ddlGender.SelectedIndex = 0;
    }
  }
}
```

Slip 7

Q.1) Write an ASP.Net program use using validation control to Validate student details form.

Solution:

StudentDetails.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="StudentDetails.aspx.cs"
Inherits="WebApplication1.StudentDetails" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Student Details Form</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Student Details Form</h2>
            <asp:Label ID="lblName" runat="server" Text="Name:"></asp:Label>
            <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="rfvName" runat="server" ControlToValidate="txtName"
ErrorMessage="Name is required." Text="*"
ValidationGroup="vgStudent"></asp:RequiredFieldValidator>
            <br />

            <asp:Label ID="lblRollNumber" runat="server" Text="Roll Number:"></asp:Label>
            <asp:TextBox ID="txtRollNumber" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="rfvRollNumber" runat="server"
ControlToValidate="txtRollNumber" ErrorMessage="Roll Number is required." Text="*"
ValidationGroup="vgStudent"></asp:RequiredFieldValidator>
            <br />

            <asp:Label ID="lblEmail" runat="server" Text="Email:"></asp:Label>
            <asp:TextBox ID="txtEmail" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="rfvEmail" runat="server" ControlToValidate="txtEmail"
ErrorMessage="Email is required." Text="*"
ValidationGroup="vgStudent"></asp:RequiredFieldValidator>
            <asp:RegularExpressionValidator ID="revEmail" runat="server"
ControlToValidate="txtEmail" ErrorMessage="Invalid email format." Text="*"
ValidationGroup="vgStudent" ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-
.]\w+)*"></asp:RegularExpressionValidator>
            <br />

            <asp:Label ID="lblMobileNumber" runat="server" Text="Mobile Number:"></asp:Label>
            <asp:TextBox ID="txtMobileNumber" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="rfvMobileNumber" runat="server"
ControlToValidate="txtMobileNumber" ErrorMessage="Mobile Number is required." Text="*"
ValidationGroup="vgStudent"></asp:RequiredFieldValidator>
            <asp:RegularExpressionValidator ID="revMobileNumber" runat="server"
ControlToValidate="txtMobileNumber" ErrorMessage="Invalid mobile number format." Text="*"
ValidationGroup="vgStudent" ValidationExpression="^[0-
9]{10}$"></asp:RegularExpressionValidator>
            <br />

            <asp:Button ID="btnSubmit" runat="server" Text="Submit" ValidationGroup="vgStudent"
OnClick="btnSubmit_Click" />
            <br />

            <asp:ValidationSummary ID="vsStudent" runat="server" ValidationGroup="vgStudent" />
        </div>
    </form>
</body>
</html>
```

StudentDetails.aspx.cs:

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

namespace WebApplication1
{
    public partial class StudentDetails : Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void btnSubmit_Click(object sender, EventArgs e)
        {
            // Submit logic
            if (Page.IsValid)
            {
                // Form is valid, process the data
                string name = txtName.Text;
                string rollNumber = txtRollNumber.Text;
                string email = txtEmail.Text;
                string mobileNumber = txtMobileNumber.Text;

                // Perform further processing or database operations
                // For demonstration purposes, display a success message
                ClientScript.RegisterStartupScript(this.GetType(), "alert", "alert('Student details submitted successfully.');", true);
            }
            else
            {
                // Form is not valid, do nothing or display an error message
            }
        }
    }
}
```

Q.2) Create an ASP.Net application which displays 5-10 quiz questions, each with four possible options and exactly one correct option; the application then counts and presents the user with the number of correct answers." and shows the result to SPPU M.Sc. Computer Science Syllabus 2023-24 of 5 user's

Solution:

Quiz.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Quiz.aspx.cs" Inherits="YourNamespace.Quiz" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Quiz</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Quiz</h2>
            <asp:PlaceHolder ID="phQuestions" runat="server"></asp:PlaceHolder>
            <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />
            <hr />
            <asp:Label ID="lblResult" runat="server" Visible="false"></asp:Label>
        </div>
    </form>
</body>
</html>
```

Quiz.aspx.cs:

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

namespace YourNamespace
```

```
{
   public partial class Quiz : System.Web.UI.Page
   {
      private List<Question> quizQuestions;

      protected void Page_Load(object sender, EventArgs e)
      {
         if (!IsPostBack)
         {
            // Initialize quiz questions (you can retrieve these from a database or XML file)
            quizQuestions = new List<Question>
            {
               new Question
               {
                  Text = "What is 2 + 2?",
                  Options = new List<string> { "3", "4", "5", "6" },
                  CorrectOptionIndex = 1 // Index of correct option (starts from 0)
               },
               // Add more questions here
            };

            // Display quiz questions dynamically
            foreach (var question in quizQuestions)
            {
               Panel panel = new Panel();
               panel.Controls.Add(new LiteralControl(question.Text + "<br />"));

               RadioButtonList rblOptions = new RadioButtonList();
               rblOptions.ID = "rblOptions_" + quizQuestions.IndexOf(question);
               rblOptions.DataSource = question.Options;
               rblOptions.DataBind();
               panel.Controls.Add(rblOptions);

               phQuestions.Controls.Add(panel);
            }
         }
      }

      protected void btnSubmit_Click(object sender, EventArgs e)
      {
         int correctCount = 0;
         for (int i = 0; i < quizQuestions.Count; i++)
         {
            RadioButtonList rblOptions = (RadioButtonList)phQuestions.FindControl("rblOptions_" +
i);
            if (rblOptions != null && rblOptions.SelectedIndex ==
quizQuestions[i].CorrectOptionIndex)
            {
               correctCount++;
            }
         }
```

```
            lblResult.Text = $"You got {correctCount} out of {quizQuestions.Count} questions correct.";
            lblResult.Visible = true;

            // Save the result to the database or display it in a report
        }
    }

    public class Question
    {
        public string Text { get; set; }
        public List<string> Options { get; set; }
        public int CorrectOptionIndex { get; set; }
    }
}
```

Slip 8

Q.1) Write web application in ASP.Net take two buttons on the page, a text box to enter string and a label to display the text stored from last session.

Solution:

Default.aspx:

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

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Session Demo</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Session Demo</h2>
            <asp:TextBox ID="txtInput" runat="server"></asp:TextBox>
            <br />
            <asp:Button ID="btnSave" runat="server" Text="Save" OnClick="btnSave_Click" />
            <asp:Button ID="btnClear" runat="server" Text="Clear" OnClick="btnClear_Click" />
            <br />
            <asp:Label ID="lblStoredText" runat="server" Text=""></asp:Label>
        </div>
    </form>
</body>
</html>
```

Default.aspx.cs:

```
using System;

namespace YourNamespace
{
    public partial class Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            // Check if text is stored in session and display it
            if (!IsPostBack)
            {
                if (Session["StoredText"] != null)
                {
                    lblStoredText.Text = Session["StoredText"].ToString();
                }
            }
        }

        protected void btnSave_Click(object sender, EventArgs e)
        {
            // Store the text entered by the user in session
            Session["StoredText"] = txtInput.Text;
            lblStoredText.Text = txtInput.Text;
        }

        protected void btnClear_Click(object sender, EventArgs e)
        {
            // Clear the stored text and label text
            Session["StoredText"] = null;
            lblStoredText.Text = "";
            txtInput.Text = "";
        }
    }
}
```

Q.2) Write an ASP.Net program to Display Students details (rollno, Name, Mobile No, Address, Gender) from database Edit, Delete information from GridView

Solution:

Default.aspx:

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

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Student Details</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Student Details</h2>
            <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
OnRowEditing="GridView1_RowEditing"
```

```
                OnRowCancelingEdit="GridView1_RowCancelingEdit"
OnRowUpdating="GridView1_RowUpdating" OnRowDeleting="GridView1_RowDeleting">
                <Columns>
                    <asp:BoundField DataField="RollNo" HeaderText="Roll No" ReadOnly="True" />
                    <asp:BoundField DataField="Name" HeaderText="Name" />
                    <asp:BoundField DataField="MobileNo" HeaderText="Mobile No" />
                    <asp:BoundField DataField="Address" HeaderText="Address" />
                    <asp:BoundField DataField="Gender" HeaderText="Gender" />
                    <asp:CommandField ShowEditButton="True" ShowDeleteButton="True" />
                </Columns>
            </asp:GridView>
        </div>
    </form>
</body>
</html>
```

Default.aspx.cs:

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

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

        private void BindGrid()
        {
            // Retrieve student details from the database and bind them to GridView
            string connectionString = "YourConnectionString";
            string query = "SELECT RollNo, Name, MobileNo, Address, Gender FROM Students";
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                using (SqlCommand command = new SqlCommand(query, connection))
                {
                    SqlDataAdapter adapter = new SqlDataAdapter(command);
                    DataTable dataTable = new DataTable();
                    adapter.Fill(dataTable);
                    GridView1.DataSource = dataTable;
                    GridView1.DataBind();
                }
            }
        }
        protected void GridView1_RowEditing(object sender, GridViewEditEventArgs e)
        {
            // Enable editing mode for the selected row
            GridView1.EditIndex = e.NewEditIndex;
            BindGrid();
        }

        protected void GridView1_RowCancelingEdit(object sender, GridViewCancelEditEventArgs e)
        {
            // Cancel editing mode
            GridView1.EditIndex = -1;
            BindGrid();
        }

        protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs e)
        {
            // Update the database with the edited values
            GridViewRow row = GridView1.Rows[e.RowIndex];
            string rollNo = ((Label)row.FindControl("lblRollNo")).Text;
            string name = ((TextBox)row.FindControl("txtName")).Text;
            string mobileNo = ((TextBox)row.FindControl("txtMobileNo")).Text;
            string address = ((TextBox)row.FindControl("txtAddress")).Text;
            string gender = ((TextBox)row.FindControl("txtGender")).Text;

            // Update the database with the edited values
            // You need to write the update query here and execute it using ADO.NET

            GridView1.EditIndex = -1;
            BindGrid();
        }

        protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
        {
            // Delete the selected row from the database
            GridViewRow row = GridView1.Rows[e.RowIndex];
            string rollNo = ((Label)row.FindControl("lblRollNo")).Text;

            // Delete the row from the database
            // You need to write the delete query here and execute it using ADO.NET

            BindGrid();
        }
    }
}
```

Slip 9

Q.1) Write a web application in asp.net to implement concept of state management using View State and query string

Solution:

Default.aspx:

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

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>State Management</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h2>State Management</h2>
      <asp:TextBox ID="txtValue" runat="server"></asp:TextBox>
      <br />
      <asp:Button ID="btnViewState" runat="server" Text="Save using View State"
OnClick="btnViewState_Click" />
      <asp:Button ID="btnQueryString" runat="server" Text="Save using Query String"
OnClick="btnQueryString_Click" />
      <br />
      <asp:Label ID="lblViewStateValue" runat="server" Text=""></asp:Label>
      <asp:Label ID="lblQueryStringValue" runat="server" Text=""></asp:Label>
    </div>
  </form>
</body>
</html>
```

Default.aspx.cs:

```
using System;

namespace YourNamespace
{
  public partial class Default : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {
      if (!IsPostBack)
      {
        // Check if value is stored in View State
        if (ViewState["ViewStateValue"] != null)
```

```
                {
                    lblViewStateValue.Text = ViewState["ViewStateValue"].ToString();
                }

                // Check if value is passed in query string
                if (!string.IsNullOrEmpty(Request.QueryString["QueryStringValue"]))
                {
                    lblQueryStringValue.Text = Request.QueryString["QueryStringValue"];
                }
            }
        }

        protected void btnViewState_Click(object sender, EventArgs e)
        {
            // Store value in View State
            ViewState["ViewStateValue"] = txtValue.Text;
            lblViewStateValue.Text = txtValue.Text;
        }

        protected void btnQueryString_Click(object sender, EventArgs e)
        {
            // Redirect to the same page with the value in query string
            Response.Redirect("Default.aspx?QueryStringValue=" + txtValue.Text);
        }
    }
}
```

Q.2) Create a web application that accepts user ID, name, password, age, and email ID. All information entry is compulsory. Password should be reconfirmed. Age should be within 21 to 30. Email ID should be valid. User ID should have at least one capital letter and digit, with a length between 7 to 20 characters.

Solution:

Registration.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Registration.aspx.cs"
Inherits="YourNamespace.Registration" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>User Registration</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>User Registration</h2>
            <asp:Label ID="lblUserId" runat="server" Text="User ID:"></asp:Label>
            <asp:TextBox ID="txtUserId" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="rfvUserId" runat="server" ControlToValidate="txtUserId"
                ErrorMessage="User ID is required"></asp:RequiredFieldValidator>
            <br />
            <asp:Label ID="lblName" runat="server" Text="Name:"></asp:Label>
            <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="rfvName" runat="server" ControlToValidate="txtName"
                ErrorMessage="Name is required"></asp:RequiredFieldValidator>
            <br />
            <asp:Label ID="lblPassword" runat="server" Text="Password:"></asp:Label>
            <asp:TextBox ID="txtPassword" runat="server" TextMode="Password"></asp:TextBox>
            <asp:RequiredFieldValidator ID="rfvPassword" runat="server"
ControlToValidate="txtPassword"
                ErrorMessage="Password is required"></asp:RequiredFieldValidator>
            <br />
            <asp:Label ID="lblConfirmPassword" runat="server" Text="Confirm
Password:"></asp:Label>
            <asp:TextBox ID="txtConfirmPassword" runat="server"
TextMode="Password"></asp:TextBox>
            <asp:RequiredFieldValidator ID="rfvConfirmPassword" runat="server"
ControlToValidate="txtConfirmPassword"
                ErrorMessage="Confirm Password is required"></asp:RequiredFieldValidator>
            <asp:CompareValidator ID="cvConfirmPassword" runat="server"
ControlToValidate="txtConfirmPassword"
                ControlToCompare="txtPassword" ErrorMessage="Passwords do not
match"></asp:CompareValidator>
            <br />
            <asp:Label ID="lblAge" runat="server" Text="Age:"></asp:Label>
            <asp:TextBox ID="txtAge" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="rfvAge" runat="server" ControlToValidate="txtAge"
                ErrorMessage="Age is required"></asp:RequiredFieldValidator>
            <asp:RangeValidator ID="rvAge" runat="server" ControlToValidate="txtAge" Type="Integer"
                MinimumValue="21" MaximumValue="30" ErrorMessage="Age must be between 21 and
30"></asp:RangeValidator>
            <br />
            <asp:Label ID="lblEmail" runat="server" Text="Email ID:"></asp:Label>
            <asp:TextBox ID="txtEmail" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="rfvEmail" runat="server" ControlToValidate="txtEmail"
                ErrorMessage="Email ID is required"></asp:RequiredFieldValidator>
            <asp:RegularExpressionValidator ID="revEmail" runat="server"
ControlToValidate="txtEmail"
                ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*"
ErrorMessage="Invalid email format"></asp:RegularExpressionValidator>
            <br />
            <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />
            <asp:Label ID="lblMessage" runat="server" Text=""></asp:Label>
        </div>
    </form>
</body>
</html>
```

Registration.aspx.cs:

```
using System;

namespace YourNamespace
```

```
{
    public partial class Registration : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            // Add additional logic if needed
        }

        protected void btnSubmit_Click(object sender, EventArgs e)
        {
            // Validate user input
            if (IsValid)
            {
                // Additional validation logic if needed

                // Display success message
                lblMessage.Text = "Registration successful!";
            }
            else
            {
                // Validation failed, display error message
                lblMessage.Text = "Please correct the errors and try again.";
            }
        }
    }
}
```

Slip 10

Q.1) Create an application which will ask the user to input his name and a message, display the two items concatenated in a label, and change the format of the label using radio buttons and check boxes for selection, the user can make the label text bold, underlined or italic and change its color . Include three buttons such as to display the message in the label, clear the text boxes and label and exit.

Solution:

Default.aspx:

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

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Message Formatting</title>
    <style>
        .bold {
            font-weight: bold;
        }
        .underline {
            text-decoration: underline;
        }
    .italic {
      font-style: italic;
    }
  </style>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h2>Message Formatting</h2>
      <asp:TextBox ID="txtName" runat="server" placeholder="Enter your name"></asp:TextBox>
      <br />
      <asp:TextBox ID="txtMessage" runat="server" placeholder="Enter your
message"></asp:TextBox>
      <br />
      <asp:RadioButtonList ID="rblFormats" runat="server">
        <asp:ListItem Text="Bold" Value="bold" />
        <asp:ListItem Text="Underline" Value="underline" />
        <asp:ListItem Text="Italic" Value="italic" />
      </asp:RadioButtonList>
      <br />
      <asp:CheckBox ID="chkColor" runat="server" Text="Change Color" />
      <br />
      <asp:Button ID="btnDisplay" runat="server" Text="Display Message"
OnClick="btnDisplay_Click" />
      <asp:Button ID="btnClear" runat="server" Text="Clear" OnClick="btnClear_Click" />
      <asp:Button ID="btnExit" runat="server" Text="Exit" OnClick="btnExit_Click" />
      <br />
      <asp:Label ID="lblDisplay" runat="server" Text=""></asp:Label>
    </div>
  </form>
</body>
</html>
```

Default.aspx.cs:

```
using System;
using System.Drawing;

namespace YourNamespace
{
  public partial class Default : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {
      // Add additional logic if needed
    }

    protected void btnDisplay_Click(object sender, EventArgs e)
    {
      string name = txtName.Text;
      string message = txtMessage.Text;
      string format = rblFormats.SelectedItem.Value;
```

```
            lblDisplay.Text = message;

            // Apply formatting based on user selection
            if (format == "bold")
            {
                lblDisplay.Font.Bold = true;
            }
            else if (format == "underline")
            {
                lblDisplay.Font.Underline = true;
            }
            else if (format == "italic")
            {
                lblDisplay.Font.Italic = true;
            }

            // Change color if checkbox is checked
            if (chkColor.Checked)
            {
                lblDisplay.ForeColor = Color.Red; // Change color to red, you can change it as per your requirement
            }
        }

        protected void btnClear_Click(object sender, EventArgs e)
        {
            // Clear text boxes and label
            txtName.Text = string.Empty;
            txtMessage.Text = string.Empty;
            lblDisplay.Text = string.Empty;
            lblDisplay.Font.Bold = false;
            lblDisplay.Font.Underline = false;
            lblDisplay.Font.Italic = false;
            lblDisplay.ForeColor = Color.Black;
        }

        protected void btnExit_Click(object sender, EventArgs e)
        {
            // Add exit logic if needed
        }
    }
}
```

Q.2) Write an ASP.NET program to accept user input for 5 teacher information into a database and Display it in a GridView

Solution:

TeacherEntry.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="TeacherEntry.aspx.cs"
Inherits="YourNamespace.TeacherEntry" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>Teacher Entry</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h2>Teacher Information Entry</h2>
      <asp:Label ID="lblName" runat="server" Text="Name:"></asp:Label>
      <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
      <br />
      <asp:Label ID="lblSubject" runat="server" Text="Subject:"></asp:Label>
      <asp:TextBox ID="txtSubject" runat="server"></asp:TextBox>
      <br />
      <asp:Label ID="lblEmail" runat="server" Text="Email:"></asp:Label>
      <asp:TextBox ID="txtEmail" runat="server"></asp:TextBox>
      <br />
      <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />
    </div>
  </form>
  <br />
  <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False">
    <Columns>
      <asp:BoundField DataField="TeacherID" HeaderText="Teacher ID" />
      <asp:BoundField DataField="Name" HeaderText="Name" />
      <asp:BoundField DataField="Subject" HeaderText="Subject" />
      <asp:BoundField DataField="Email" HeaderText="Email" />
    </Columns>
  </asp:GridView>
</body>
</html>
```

TeacherEntry.aspx.cs:

```
using System;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;

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

    protected void btnSubmit_Click(object sender, EventArgs e)
    {
      string name = txtName.Text;
      string subject = txtSubject.Text;
      string email = txtEmail.Text;

      // Insert data into the database
      InsertTeacherInfo(name, subject, email);

      // Re-bind GridView to display updated data
      BindGrid();

      // Clear input fields after submission
      ClearFields();
    }

    private void InsertTeacherInfo(string name, string subject, string email)
    {
      string connectionString =
ConfigurationManager.ConnectionStrings["YourConnectionString"].ConnectionString;
      using (SqlConnection connection = new SqlConnection(connectionString))
      {
        string query = "INSERT INTO Teachers (Name, Subject, Email) VALUES (@Name,
@Subject, @Email)";
        using (SqlCommand command = new SqlCommand(query, connection))
        {
          command.Parameters.AddWithValue("@Name", name);
          command.Parameters.AddWithValue("@Subject", subject);
          command.Parameters.AddWithValue("@Email", email);
          connection.Open();
          command.ExecuteNonQuery();
        }
      }
    }

    private void BindGrid()
    {
      string connectionString =
ConfigurationManager.ConnectionStrings["YourConnectionString"].ConnectionString;
      string query = "SELECT TeacherID, Name, Subject, Email FROM Teachers";
      using (SqlConnection connection = new SqlConnection(connectionString))
```

```
            {
                using (SqlCommand command = new SqlCommand(query, connection))
                {
                    SqlDataAdapter adapter = new SqlDataAdapter(command);
                    DataTable dataTable = new DataTable();
                    adapter.Fill(dataTable);
                    GridView1.DataSource = dataTable;
                    GridView1.DataBind();
                }
            }
        }

        private void ClearFields()
        {
            txtName.Text = string.Empty;
            txtSubject.Text = string.Empty;
            txtEmail.Text = string.Empty;
        }
    }
}
```

Slip 11

Q.1)Write web application in ASP.NET take two button on the pages, a text box to enter string and label to display the text stored from last session.

<u>Solution:</u>

Default.aspx:

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

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Session Example</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Session Example</h2>
            <asp:TextBox ID="txtInput" runat="server" placeholder="Enter text"></asp:TextBox>
            <br />
            <asp:Button ID="btnSave" runat="server" Text="Save" OnClick="btnSave_Click" />
            <asp:Button ID="btnClear" runat="server" Text="Clear" OnClick="btnClear_Click" />
            <br />
            <asp:Label ID="lblStoredText" runat="server" Text=""></asp:Label>
        </div>
    </form>
</body>
</html>
```

Default.aspx.cs:

```
using System;

namespace YourNamespace
{
    public partial class Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                // Check if session variable exists and display its value
                if (Session["StoredText"] != null)
                {
                    lblStoredText.Text = Session["StoredText"].ToString();
                }
            }
        }

        protected void btnSave_Click(object sender, EventArgs e)
        {
            // Store text from textbox in session variable
            Session["StoredText"] = txtInput.Text;

            // Display stored text in label
            lblStoredText.Text = txtInput.Text;
        }
        protected void btnClear_Click(object sender, EventArgs e)
        {
            // Clear session variable and label text
            Session["StoredText"] = null;
            lblStoredText.Text = "";
        }
    }
}
```

Q.2) Write an ASP.Net program to Display Employee details (EmpID, Name, Designation, Joining Date, Mob.no, Gender) from database Edit, Delete information from Grid View.

Solution:

EmployeeDetails.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="EmployeeDetails.aspx.cs"
Inherits="YourNamespace.EmployeeDetails" %>
```

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Employee Details</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <h2>Employee Details</h2>
         <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
OnRowEditing="GridView1_RowEditing" OnRowUpdating="GridView1_RowUpdating"
OnRowCancelingEdit="GridView1_RowCancelingEdit"
OnRowDeleting="GridView1_RowDeleting" DataKeyNames="EmpID">
            <Columns>
               <asp:BoundField DataField="EmpID" HeaderText="EmpID" ReadOnly="True" />
               <asp:BoundField DataField="Name" HeaderText="Name" />
               <asp:BoundField DataField="Designation" HeaderText="Designation" />
               <asp:BoundField DataField="JoiningDate" HeaderText="Joining Date"
DataFormatString="{0:dd/MM/yyyy}" />
               <asp:BoundField DataField="MobileNumber" HeaderText="Mobile Number" />
               <asp:BoundField DataField="Gender" HeaderText="Gender" />
               <asp:CommandField ShowEditButton="True" />
               <asp:CommandField ShowDeleteButton="True" />
            </Columns>
         </asp:GridView>
      </div>
   </form>
</body>
</html>
```

EmployeeDetails.aspx.cs:

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

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

```
protected void BindGridView()
{
    string connectionString = ConfigurationManager.ConnectionStrings["YourConnectionString"].ConnectionString;
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        string query = "SELECT EmpID, Name, Designation, JoiningDate, MobileNumber, Gender FROM Employees";
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            SqlDataAdapter adapter = new SqlDataAdapter(command);
            DataTable dataTable = new DataTable();
            adapter.Fill(dataTable);
            GridView1.DataSource = dataTable;
            GridView1.DataBind();
        }
    }
}

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

protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs e)
{
    GridViewRow row = GridView1.Rows[e.RowIndex];
    int empID = Convert.ToInt32(GridView1.DataKeys[e.RowIndex].Value);
    string name = ((TextBox)row.FindControl("txtName")).Text;
    string designation = ((TextBox)row.FindControl("txtDesignation")).Text;
    // Retrieve and parse other updated values as needed
    // Update the database with the new values
    // Example: UpdateEmployee(empID, name, designation);
    GridView1.EditIndex = -1;
    BindGridView();
}

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

protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
{
    int empID = Convert.ToInt32(GridView1.DataKeys[e.RowIndex].Value);
    // Delete employee from database
    // Example: DeleteEmployee(empID);
    BindGridView();
}
```

```
    }
}
```

Slip 12

Q.1) Write a web application in asp.net using c# to accept the user's information and on click on submit button entered information should get displayed into next page(session).

<u>Solution:</u>

InputPage.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="InputPage.aspx.cs"
Inherits="YourNamespace.InputPage" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Input Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Enter User Information</h2>
            <asp:TextBox ID="txtName" runat="server" placeholder="Enter Name"></asp:TextBox>
            <br />
            <asp:TextBox ID="txtEmail" runat="server" placeholder="Enter Email"></asp:TextBox>
            <br />
            <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />
        </div>
    </form>
</body>
</html>
```

DisplayPage.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="DisplayPage.aspx.cs"
Inherits="YourNamespace.DisplayPage" %>
```

```
<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Display Page</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <h2>User Information</h2>
         <asp:Label ID="lblName" runat="server"></asp:Label>
         <br />
         <asp:Label ID="lblEmail" runat="server"></asp:Label>
      </div>
   </form>
</body>
</html>
```

DisplayPage.aspx.cs:

```
using System;

namespace YourNamespace
{
   public partial class DisplayPage : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         // Retrieve and display stored session variables
         if (Session["Name"] != null)
         {
            lblName.Text = "Name: " + Session["Name"].ToString();
         }
         if (Session["Email"] != null)
         {
            lblEmail.Text = "Email: " + Session["Email"].ToString();
         }
      }
   }
}
```

Q.2) Write an ASP.Net program to Display Teacher details (Teacher_ID, Name,Designation, Joining Date, Mob.no, Gender) from database Edit, Delete information from Grid View.

Solution:

TeacherDetails.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="TeacherDetails.aspx.cs"
Inherits="YourNamespace.TeacherDetails" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
    <title>Teacher Details</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Teacher Details</h2>
            <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
OnRowEditing="GridView1_RowEditing" OnRowUpdating="GridView1_RowUpdating"
OnRowCancelingEdit="GridView1_RowCancelingEdit"
OnRowDeleting="GridView1_RowDeleting" DataKeyNames="Teacher_ID">
                <Columns>
                    <asp:BoundField DataField="Teacher_ID" HeaderText="Teacher ID" ReadOnly="True"
/>
                    <asp:BoundField DataField="Name" HeaderText="Name" />
                    <asp:BoundField DataField="Designation" HeaderText="Designation" />
                    <asp:BoundField DataField="JoiningDate" HeaderText="Joining Date"
DataFormatString="{0:dd/MM/yyyy}" />
                    <asp:BoundField DataField="MobileNumber" HeaderText="Mobile Number" />
                    <asp:BoundField DataField="Gender" HeaderText="Gender" />
                    <asp:CommandField ShowEditButton="True" />
                    <asp:CommandField ShowDeleteButton="True" />
                </Columns>
            </asp:GridView>
        </div>
    </form>
</body>
</html>
```

TeacherDetails.aspx.cs:

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

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

        protected void BindGridView()
        {
            string connectionString =
ConfigurationManager.ConnectionStrings["YourConnectionString"].ConnectionString;
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                string query = "SELECT Teacher_ID, Name, Designation, JoiningDate, MobileNumber,
Gender FROM Teachers";
                using (SqlCommand command = new SqlCommand(query, connection))
                {
                    SqlDataAdapter adapter = new SqlDataAdapter(command);
                    DataTable dataTable = new DataTable();
                    adapter.Fill(dataTable);
                    GridView1.DataSource = dataTable;
                    GridView1.DataBind();
                }
            }
        }

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

        protected void GridView1_RowUpdating(object sender, GridViewUpdateEventArgs e)
        {
            GridViewRow row = GridView1.Rows[e.RowIndex];
            int teacherID = Convert.ToInt32(GridView1.DataKeys[e.RowIndex].Value);
            string name = ((TextBox)row.FindControl("txtName")).Text;
            string designation = ((TextBox)row.FindControl("txtDesignation")).Text;
            // Retrieve and parse other updated values as needed
            // Update the database with the new values
            // Example: UpdateTeacher(teacherID, name, designation);
            GridView1.EditIndex = -1;
            BindGridView();
        }

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

        protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
        {
            int teacherID = Convert.ToInt32(GridView1.DataKeys[e.RowIndex].Value);
            // Delete teacher from database
            // Example: DeleteTeacher(teacherID);
            BindGridView();
        }
    }
}
```

Slip 13

Q.1) Write a program that uses a textbox for a user input name and validate it for RequiredField Validation

Solution:

Default.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="WebApplication1.Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>RequiredFieldValidation Example</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>User Registration</h2>
            <asp:Label ID="lblName" runat="server" Text="Name:"></asp:Label>
            <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="rfvName" runat="server" ControlToValidate="txtName"
ErrorMessage="Name is required" Text="*"></asp:RequiredFieldValidator>
            <br />
            <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />
            <br />
            <asp:Label ID="lblMessage" runat="server" ForeColor="Red" Visible="false"></asp:Label>
        </div>
    </form>
</body>
</html>
```

Default.aspx.cs:

```
using System;

namespace WebApplication1
{
    public partial class Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }

        protected void btnSubmit_Click(object sender, EventArgs e)
        {
            if (Page.IsValid)
            {
                string name = txtName.Text;
                lblMessage.Text = "Hello, " + name + "! Registration successful!";
                lblMessage.Visible = true;
            }
            else
            {
                lblMessage.Text = "Please fill in the required fields.";
                lblMessage.Visible = true;
            }
        }
    }
}
```

Q.2) Write an ASP.net program, the user can enter 5 employee’s information in database and display into the gridview.

Solution:

EmployeeForm.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="EmployeeForm.aspx.cs"
Inherits="YourNamespace.EmployeeForm" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Employee Form</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Enter Employee Information</h2>
            <asp:Label ID="lblName" runat="server" Text="Name:"></asp:Label>
            <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
            <br />
            <asp:Label ID="lblDesignation" runat="server" Text="Designation:"></asp:Label>
```

```
      <asp:TextBox ID="txtDesignation" runat="server"></asp:TextBox>
      <br />
      <!-- Add more input fields for other employee information -->
      <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />
    </div>
  </form>
</body>
</html>
```

EmployeeForm.aspx.cs:

```
using System;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;

namespace YourNamespace
{
  public partial class EmployeeForm : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void btnSubmit_Click(object sender, EventArgs e)
    {
      string name = txtName.Text;
      string designation = txtDesignation.Text;
      // Retrieve other employee information from input fields

      // Insert employee information into database
      InsertEmployee(name, designation);

      // Redirect to the page showing all employees
      Response.Redirect("EmployeeList.aspx");
    }

    protected void InsertEmployee(string name, string designation)
    {
      string connectionString =
ConfigurationManager.ConnectionStrings["YourConnectionString"].ConnectionString;
      using (SqlConnection connection = new SqlConnection(connectionString))
      {
        string query = "INSERT INTO Employees (Name, Designation) VALUES (@Name,
@Designation)";
        using (SqlCommand command = new SqlCommand(query, connection))
        {
          command.Parameters.AddWithValue("@Name", name);
          command.Parameters.AddWithValue("@Designation", designation);
          // Add parameters for other employee information
```

```
                    connection.Open();
                    command.ExecuteNonQuery();
                }
            }
        }
    }
}
```

EmployeeList.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="EmployeeList.aspx.cs"
Inherits="YourNamespace.EmployeeList" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Employee List</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Employee List</h2>
            <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False">
                <Columns>
                    <asp:BoundField DataField="Name" HeaderText="Name" />
                    <asp:BoundField DataField="Designation" HeaderText="Designation" />
                    <!-- Add more BoundFields for other employee information -->
                </Columns>
            </asp:GridView>
        </div>
    </form>
</body>
</html>
```

EmployeeList.aspx.cs:

```
using System;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;

namespace YourNamespace
{
    public partial class EmployeeList : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                BindGridView();
            }
        }
        protected void BindGridView()
        {
            string connectionString =
ConfigurationManager.ConnectionStrings["YourConnectionString"].ConnectionString;
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                string query = "SELECT Name, Designation FROM Employees";
                using (SqlCommand command = new SqlCommand(query, connection))
                {
                    SqlDataAdapter adapter = new SqlDataAdapter(command);
                    DataTable dataTable = new DataTable();
                    adapter.Fill(dataTable);
                    GridView1.DataSource = dataTable;
                    GridView1.DataBind();
                }
            }
        }
    }
}
```

Slip 14

Q.1) Write a program that uses a textbox for a user input name and validate it for Compared Validation.

Solution:

Default.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.aspx.cs"
Inherits="WebApplication1.Default" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
  <title>Compared Validation Example</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h2>User Registration</h2>
      <asp:Label ID="lblPassword" runat="server" Text="Password:"></asp:Label>
      <asp:TextBox ID="txtPassword" runat="server" TextMode="Password"></asp:TextBox>
      <br />
      <asp:Label ID="lblConfirmPassword" runat="server" Text="Confirm
Password:"></asp:Label>
      <asp:TextBox ID="txtConfirmPassword" runat="server"
TextMode="Password"></asp:TextBox>
      <asp:CompareValidator ID="cvConfirmPassword" runat="server"
ControlToValidate="txtConfirmPassword" ControlToCompare="txtPassword"
ErrorMessage="Passwords do not match" Text="*"></asp:CompareValidator>
      <br />
      <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />
      <br />
      <asp:Label ID="lblMessage" runat="server" ForeColor="Red" Visible="false"></asp:Label>
    </div>
  </form>
</body>
</html>
```

Default.aspx.cs:

```
using System;

namespace WebApplication1
{
  public partial class Default : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void btnSubmit_Click(object sender, EventArgs e)
    {
      if (Page.IsValid)
      {
        // Registration successful
        lblMessage.Text = "Registration successful!";
        lblMessage.Visible = true;
      }
      else
      {
        // Validation failed
        lblMessage.Text = "Please correct the errors.";
        lblMessage.Visible = true;
```

```
            }
        }
    }
}
```

Q.2) Create an ASP.Net application which displays 5-10 quiz questions, each with four possible options and exactly one correct option; the application then counts and presents the user with the number of correct answers." and shows the result to SPPU M.S.C.(ComputerScience) Syllabus 2023-24 of 5 users.

Solution:

QuizPage.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="QuizPage.aspx.cs"
Inherits="YourNamespace.QuizPage" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Quiz</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Quiz</h2>
            <asp:Repeater ID="rptQuestions" runat="server">
                <ItemTemplate>
                    <div>
                        <asp:Label ID="lblQuestion" runat="server" Text='<%# Eval("QuestionText")
%>'></asp:Label>
                        <br />
                        <asp:RadioButtonList ID="rblOptions" runat="server" RepeatDirection="Vertical"
DataSource='<%# Eval("Options") %>' DataTextField="OptionText"
DataValueField="OptionId"></asp:RadioButtonList>
                        <br />
                    </div>
                </ItemTemplate>
            </asp:Repeater>
            <br />
            <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />
        </div>
    </form>
</body>
</html>
```

QuizPage.aspx.cs:

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

namespace YourNamespace
{
    public partial class QuizPage : System.Web.UI.Page
    {
        // Mock data for quiz questions and options (replace with actual data source)
        private List<Question> quizQuestions = new List<Question>
        {
            new Question
            {
                QuestionText = "What is the capital of France?",
                Options = new List<Option>
                {
                    new Option { OptionId = 1, OptionText = "London" },
                    new Option { OptionId = 2, OptionText = "Paris", IsCorrect = true },
                    new Option { OptionId = 3, OptionText = "Berlin" },
                    new Option { OptionId = 4, OptionText = "Rome" }
                }
            },
            // Add more questions here
        };

        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                rptQuestions.DataSource = quizQuestions;
                rptQuestions.DataBind();
            }
        }

        protected void btnSubmit_Click(object sender, EventArgs e)
        {
            // Calculate number of correct answers for each user and display results
            // Logic for scoring and displaying results goes here
        }
    }

    public class Question
    {
        public string QuestionText { get; set; }
        public List<Option> Options { get; set; }
    }

    public class Option
    {
        public int OptionId { get; set; }
        public string OptionText { get; set; }
        public bool IsCorrect { get; set; }
    }
}
```

Slip 15

Q.1) Write a web application in asp.net to implement concept of state management using View State and query string.

Soluttion:

ViewStateQueryStringExample.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="ViewStateQueryStringExample.aspx.cs"
Inherits="YourNamespace.ViewStateQueryStringExample" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>View State and Query String Example</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h2>State Management Example</h2>
      <asp:TextBox ID="txtData" runat="server"></asp:TextBox>
      <br />
      <asp:Button ID="btnSaveViewState" runat="server" Text="Save to View State"
OnClick="btnSaveViewState_Click" />
      <asp:Button ID="btnRedirect" runat="server" Text="Redirect with Query String"
OnClick="btnRedirect_Click" />
    </div>
  </form>
</body>
</html>
```

ViewStateQueryStringExample.aspx.cs:

```
using System;

namespace YourNamespace
{
  public partial class ViewStateQueryStringExample : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {
      if (!IsPostBack)
      {
        // Check if there's data stored in View State
        if (ViewState["Data"] != null)
        {
          txtData.Text = ViewState["Data"].ToString();
        }
      }
    }

    protected void btnSaveViewState_Click(object sender, EventArgs e)
    {
            // Save data to View State
            ViewState["Data"] = txtData.Text;
        }

        protected void btnRedirect_Click(object sender, EventArgs e)
        {
            // Redirect to another page with data in query string
            string data = txtData.Text;
            Response.Redirect("NextPage.aspx?data=" + data);
        }
    }
}
```

NextPage.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="NextPage.aspx.cs"
Inherits="YourNamespace.NextPage" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Next Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Next Page</h2>
            <asp:Label ID="lblData" runat="server" Text=""></asp:Label>
        </div>
    </form>
</body>
</html>
```

NextPage.aspx.cs:

```
using System;

namespace YourNamespace
{
    public partial class NextPage : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                // Retrieve data from query string
                if (Request.QueryString["data"] != null)
                {
                    string data = Request.QueryString["data"];
                    lblData.Text = "Data received from query string: " + data;
                }
            }
        }
    }
}
```

Q.2)Write an ASP.Net program of Student Registration form by using validation control(Range Validator, Field Validator)

Solution:

StudentRegistrationForm.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="StudentRegistrationForm.aspx.cs"
Inherits="YourNamespace.StudentRegistrationForm" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Student Registration Form</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Student Registration Form</h2>
            <asp:Label ID="lblName" runat="server" Text="Name:"></asp:Label>
            <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="rfvName" runat="server" ControlToValidate="txtName"
ErrorMessage="Name is required" Text="*"></asp:RequiredFieldValidator>
            <br />
            <asp:Label ID="lblAge" runat="server" Text="Age:"></asp:Label>
            <asp:TextBox ID="txtAge" runat="server"></asp:TextBox>
            <asp:RangeValidator ID="rvAge" runat="server" ControlToValidate="txtAge"
ErrorMessage="Age must be between 18 and 30" MinimumValue="18" MaximumValue="30"
Type="Integer" Text="*"></asp:RangeValidator>
            <br />
            <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />
            <br />
            <asp:Label ID="lblMessage" runat="server" ForeColor="Red" Visible="false"></asp:Label>
        </div>
    </form>
</body>
</html>
```

StudentRegistrationForm.aspx.cs:

```
using System;

namespace YourNamespace
{
    public partial class StudentRegistrationForm : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }

        protected void btnSubmit_Click(object sender, EventArgs e)
        {
            if (Page.IsValid)
            {
                // Registration successful
                lblMessage.Text = "Registration successful!";
                lblMessage.Visible = true;
            }
            else
            {
                // Validation failed
                lblMessage.Text = "Please correct the errors.";
                lblMessage.Visible = true;
            }
        }
    }
}
```

Slip 16

Q.1) Write a web application in asp.net using c# to accept the user's information and on click on submit button entered information should get displayed into next page(session).

Solution:

UserInformationForm.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="UserInformationForm.aspx.cs"
Inherits="YourNamespace.UserInformationForm" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>User Information Form</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>User Information Form</h2>
            <asp:Label ID="lblName" runat="server" Text="Name:"></asp:Label>
            <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
            <br />
            <asp:Label ID="lblEmail" runat="server" Text="Email:"></asp:Label>
            <asp:TextBox ID="txtEmail" runat="server"></asp:TextBox>
            <br />
            <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />
        </div>
    </form>
</body>
</html>
```

UserInformationForm.aspx.cs:

```
using System;
```

```
namespace YourNamespace
{
   public partial class UserInformationForm : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {

      }

      protected void btnSubmit_Click(object sender, EventArgs e)
      {
         // Store user information in session
         Session["Name"] = txtName.Text;
         Session["Email"] = txtEmail.Text;

         // Redirect to the next page
         Response.Redirect("NextPage.aspx");
      }
   }
}
```

NextPage.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="NextPage.aspx.cs"
Inherits="YourNamespace.NextPage" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Next Page</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <h2>User Information</h2>
         <asp:Label ID="lblName" runat="server"></asp:Label>
         <br />
         <asp:Label ID="lblEmail" runat="server"></asp:Label>
      </div>
   </form>
</body>
</html>
```

NextPage.aspx.cs:

```
using System;

namespace YourNamespace
{
   public partial class NextPage : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
            if (!IsPostBack)
            {
                // Retrieve user information from session and display it
                if (Session["Name"] != null)
                {
                    lblName.Text = "Name: " + Session["Name"].ToString();
                }

                if (Session["Email"] != null)
                {
                    lblEmail.Text = "Email: " + Session["Email"].ToString();
                }
            }
        }
    }
}
```

Q.2) Write an ASP.Net program to Display Product details (Product _ID, Name,Price, Quantity) from database Edit, Delete information from Grid View.

Solution:

ProductDetails.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="ProductDetails.aspx.cs"
Inherits="YourNamespace.ProductDetails" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Product Details</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Product Details</h2>
            <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
OnRowEditing="GridView1_RowEditing" OnRowDeleting="GridView1_RowDeleting">
                <Columns>
                    <asp:BoundField DataField="Product_ID" HeaderText="Product ID" ReadOnly="True"
/>
                    <asp:BoundField DataField="Name" HeaderText="Name" />
                    <asp:BoundField DataField="Price" HeaderText="Price" />
                    <asp:BoundField DataField="Quantity" HeaderText="Quantity" />
                    <asp:CommandField ShowEditButton="True" ButtonType="Button" />
                    <asp:CommandField ShowDeleteButton="True" ButtonType="Button" />
                </Columns>
            </asp:GridView>
        </div>
    </form>
</body>
</html>
```

ProductDetails.aspx.cs:

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

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

      protected void BindGridView()
      {
         string connectionString = "YourConnectionString";
         string query = "SELECT Product_ID, Name, Price, Quantity FROM Products";

         using (SqlConnection connection = new SqlConnection(connectionString))
         {
            using (SqlCommand command = new SqlCommand(query, connection))
            {
               SqlDataAdapter adapter = new SqlDataAdapter(command);
               DataTable dataTable = new DataTable();
               adapter.Fill(dataTable);
               GridView1.DataSource = dataTable;
               GridView1.DataBind();
            }
         }
      }

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

      protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
      {
         string productId = GridView1.DataKeys[e.RowIndex].Values["Product_ID"].ToString();
         string connectionString = "YourConnectionString";
         string query = "DELETE FROM Products WHERE Product_ID = @Product_ID";

         using (SqlConnection connection = new SqlConnection(connectionString))
         {
            using (SqlCommand command = new SqlCommand(query, connection))
            {
               command.Parameters.AddWithValue("@Product_ID", productId);
```

```
                connection.Open();
                command.ExecuteNonQuery();
            }
        }

        BindGridView();
    }
  }
}
```

Slip 17

Q.1) Write an ASP .net program to display list of all the files in a directory on a web form. The filesmust be displayed in a grid view control.

<u>Solution:</u>

FileList.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="FileList.aspx.cs"
Inherits="YourNamespace.FileList" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>File List</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h2>File List</h2>
      <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="false">
        <Columns>
          <asp:BoundField DataField="FileName" HeaderText="File Name" />
          <asp:BoundField DataField="FileSize" HeaderText="File Size (KB)" />
          <asp:BoundField DataField="CreationTime" HeaderText="Creation Time" />
        </Columns>
      </asp:GridView>
    </div>
  </form>
</body>
</html>
```

FileList.aspx.cs:

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

namespace YourNamespace
{
  public partial class FileList : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {
            if (!IsPostBack)
            {
                // Directory path to fetch files
                string directoryPath = Server.MapPath("~/Files"); // Change "Files" to your directory path

                // Check if directory exists
                if (Directory.Exists(directoryPath))
                {
                    // Get list of files
                    string[] files = Directory.GetFiles(directoryPath);

                    // Create DataTable to hold file details
                    System.Data.DataTable dt = new System.Data.DataTable();
                    dt.Columns.AddRange(new DataColumn[3] { new DataColumn("FileName"), new
DataColumn("FileSize"), new DataColumn("CreationTime") });

                    // Iterate through each file and add details to DataTable
                    foreach (string filePath in files)
                    {
                        FileInfo fileInfo = new FileInfo(filePath);
                        dt.Rows.Add(fileInfo.Name, fileInfo.Length / 1024, fileInfo.CreationTime);
                    }

                    // Bind DataTable to GridView
                    GridView1.DataSource = dt;
                    GridView1.DataBind();
                }
                else
                {
                    // Display error message if directory doesn't exist
                    Response.Write("Directory not found.");
                }
            }
        }
    }
}
```

Q.2) Write an ASP.Net program to Display Employee details (EmpID, Name, Designation, JoiningDate, Mob.no, Gender) from database Edit, Delete information from Grid View

Solution:

EmployeeDetails.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="EmployeeDetails.aspx.cs"
Inherits="YourNamespace.EmployeeDetails" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Employee Details</title>
</head>
```

```
<body>
  <form id="form1" runat="server">
    <div>
      <h2>Employee Details</h2>
      <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
OnRowEditing="GridView1_RowEditing" OnRowDeleting="GridView1_RowDeleting">
        <Columns>
          <asp:BoundField DataField="EmpID" HeaderText="EmpID" ReadOnly="True" />
          <asp:BoundField DataField="Name" HeaderText="Name" />
          <asp:BoundField DataField="Designation" HeaderText="Designation" />
          <asp:BoundField DataField="JoiningDate" HeaderText="Joining Date" />
          <asp:BoundField DataField="MobNo" HeaderText="Mob. No." />
          <asp:BoundField DataField="Gender" HeaderText="Gender" />
          <asp:CommandField ShowEditButton="True" ButtonType="Button" />
          <asp:CommandField ShowDeleteButton="True" ButtonType="Button" />
        </Columns>
      </asp:GridView>
    </div>
  </form>
</body>
</html>
```

EmployeeDetails.aspx.cs:

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

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

    protected void BindGridView()
    {
      string connectionString = "YourConnectionString";
      string query = "SELECT EmpID, Name, Designation, JoiningDate, MobNo, Gender FROM
Employees";

      using (SqlConnection connection = new SqlConnection(connectionString))
      {
        using (SqlCommand command = new SqlCommand(query, connection))
        {
          SqlDataAdapter adapter = new SqlDataAdapter(command);
          DataTable dataTable = new DataTable();
```

```
                adapter.Fill(dataTable);
                GridView1.DataSource = dataTable;
                GridView1.DataBind();
            }
        }
    }

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

    protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
    {
        string empId = GridView1.DataKeys[e.RowIndex].Values["EmpID"].ToString();
        string connectionString = "YourConnectionString";
        string query = "DELETE FROM Employees WHERE EmpID = @EmpID";

        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            using (SqlCommand command = new SqlCommand(query, connection))
            {
                command.Parameters.AddWithValue("@EmpID", empId);
                connection.Open();
                command.ExecuteNonQuery();
            }
        }

        BindGridView();
    }
  }
}
```

Slip 18

Q.1)Write an ASP.Net program of Student Registration form by using validation control(Range Validator, Field Validator)

Solution:

StudentRegistration.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="StudentRegistration.aspx.cs"
Inherits="YourNamespace.StudentRegistration" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Student Registration</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <h2>Student Registration</h2>
         <asp:Label ID="lblName" runat="server" Text="Name:"></asp:Label>
         <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
         <asp:RequiredFieldValidator ID="rfvName" runat="server" ControlToValidate="txtName"
ErrorMessage="Name is required" Display="Dynamic" />
         <br />
         <asp:Label ID="lblAge" runat="server" Text="Age:"></asp:Label>
         <asp:TextBox ID="txtAge" runat="server"></asp:TextBox>
         <asp:RangeValidator ID="rvAge" runat="server" ControlToValidate="txtAge"
ErrorMessage="Age must be between 18 and 25" MinimumValue="18" MaximumValue="25"
Type="Integer" Display="Dynamic" />
         <br />
         <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />
         <br />
         <asp:Label ID="lblMessage" runat="server"></asp:Label>
      </div>
   </form>
</body>
</html>
```

StudentRegistration.aspx.cs:

```
using System;

namespace YourNamespace
{
   public partial class StudentRegistration : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {

      }

      protected void btnSubmit_Click(object sender, EventArgs e)
      {
         if (IsValid)
         {
                // Process form submission
                string name = txtName.Text;
                int age = Convert.ToInt32(txtAge.Text);

                // Display message
                lblMessage.Text = "Registration successful. Name: " + name + ", Age: " + age;
            }
            else
            {
                // Validation failed, display error message
                lblMessage.Text = "Please fix validation errors.";
            }
        }
    }
}
```

Q.2) Write an ASP.Net program to Display Product details (Product _ID, Name,Price,Quantity) from database Edit, Delete information from Grid View.

<u>Solution:</u>

ProductDetails.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="ProductDetails.aspx.cs"
Inherits="YourNamespace.ProductDetails" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Product Details</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Product Details</h2>
            <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
OnRowEditing="GridView1_RowEditing" OnRowDeleting="GridView1_RowDeleting">
                <Columns>
                    <asp:BoundField DataField="Product_ID" HeaderText="Product ID" ReadOnly="True"
/>
                    <asp:BoundField DataField="Name" HeaderText="Name" />
                    <asp:BoundField DataField="Price" HeaderText="Price" />
                    <asp:BoundField DataField="Quantity" HeaderText="Quantity" />
                    <asp:CommandField ShowEditButton="True" ButtonType="Button" />
                    <asp:CommandField ShowDeleteButton="True" ButtonType="Button" />
                </Columns>
            </asp:GridView>
        </div>
    </form>
</body>
</html>
```

ProductDetails.aspx.cs:

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

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

        protected void BindGridView()
        {
            string connectionString = "YourConnectionString";
            string query = "SELECT Product_ID, Name, Price, Quantity FROM Products";

            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                using (SqlCommand command = new SqlCommand(query, connection))
                {
                    SqlDataAdapter adapter = new SqlDataAdapter(command);
                    DataTable dataTable = new DataTable();
                    adapter.Fill(dataTable);
                    GridView1.DataSource = dataTable;
                    GridView1.DataBind();
                }
            }
        }

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

        protected void GridView1_RowDeleting(object sender, GridViewDeleteEventArgs e)
        {
            string productId = GridView1.DataKeys[e.RowIndex].Values["Product_ID"].ToString();
            string connectionString = "YourConnectionString";
            string query = "DELETE FROM Products WHERE Product_ID = @Product_ID";

            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                using (SqlCommand command = new SqlCommand(query, connection))
                {
```

```
                command.Parameters.AddWithValue("@Product_ID", productId);
                connection.Open();
                command.ExecuteNonQuery();
            }
        }

        BindGridView();
    }
  }
}
```

Slip 19

Q.1) Write a web application in asp.net using c# to accept the user's information and on click on submit button entered information should get displayed into next page(session).

<u>Solution:</u>

UserInfo.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="UserInfo.aspx.cs"
Inherits="YourNamespace.UserInfo" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>User Information</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h2>User Information</h2>
      <asp:Label ID="lblName" runat="server" Text="Name:"></asp:Label>
      <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
      <br />
      <asp:Label ID="lblEmail" runat="server" Text="Email:"></asp:Label>
      <asp:TextBox ID="txtEmail" runat="server"></asp:TextBox>
      <br />
      <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />
    </div>
  </form>
</body>
</html>
```

UserInfo.aspx.cs:

```
using System;

namespace YourNamespace
{
  public partial class UserInfo : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void btnSubmit_Click(object sender, EventArgs e)
    {
        // Store user information in session
        Session["UserName"] = txtName.Text;
        Session["UserEmail"] = txtEmail.Text;

        // Redirect to the next page
        Response.Redirect("DisplayUserInfo.aspx");
    }
  }
}
```

DisplayUserInfo.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="DisplayUserInfo.aspx.cs"
Inherits="YourNamespace.DisplayUserInfo" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Display User Information</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>User Information</h2>
            <asp:Label ID="lblDisplayName" runat="server"></asp:Label>
            <br />
            <asp:Label ID="lblDisplayEmail" runat="server"></asp:Label>
        </div>
    </form>
</body>
</html>
```

DisplayUserInfo.aspx.cs:

```
using System;

namespace YourNamespace
{
    public partial class DisplayUserInfo : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                // Retrieve user information from session and display it
                if (Session["UserName"] != null)
                {
                    lblDisplayName.Text = "Name: " + Session["UserName"].ToString();
                }
                if (Session["UserEmail"] != null)
                {
                    lblDisplayEmail.Text = "Email: " + Session["UserEmail"].ToString();
                }
            }
        }
    }
}
```

Q.2) Create an ASP.Net application, which show to the user 5-10 quiz questions. All questions have 4 possible options and one right option exactly. Application counts andshows to the user how many right answers were right and shows the result to 5 users.

Solution:

QuizQuestions.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="QuizQuestions.aspx.cs"
Inherits="YourNamespace.QuizQuestions" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Quiz Questions</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Quiz Questions</h2>
            <!-- Display quiz questions and options here -->
            <asp:Literal ID="litQuestion1" runat="server"></asp:Literal>
            <asp:RadioButtonList ID="rbList1" runat="server"></asp:RadioButtonList>
            <!-- Repeat for other questions -->
            <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />
        </div>
    </form>
</body>
</html>
```

QuizResult.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="QuizResult.aspx.cs"
Inherits="YourNamespace.QuizResult" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Quiz Result</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Quiz Result</h2>
```

```
         <asp:Label ID="lblResult" runat="server" Text="Your score: "></asp:Label>
         <asp:Label ID="lblScore" runat="server"></asp:Label>
      </div>
   </form>
</body>
</html>
```

QuizResult.aspx.cs:

```
using System;

namespace YourNamespace
{
   public partial class QuizResult : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (!IsPostBack)
         {
            // Display the user's score
            string score = Request.QueryString["score"];
            lblScore.Text = score;
         }
      }
   }
}
```

Slip 20

Q.1)Write web application in ASP.Net take two buttons on the page, a text box to enter string and a label to display the text stored from last session.

Solution:

Default.aspx:

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

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Session Example</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <h2>Session Example</h2>
         <asp:TextBox ID="txtInput" runat="server"></asp:TextBox>
         <br />
         <asp:Button ID="btnSave" runat="server" Text="Save" OnClick="btnSave_Click" />
         <asp:Button ID="btnRetrieve" runat="server" Text="Retrieve" OnClick="btnRetrieve_Click"
/>
         <br />
         <asp:Label ID="lblStoredText" runat="server"></asp:Label>
```

```
        </div>
    </form>
</body>
</html>
```

Default.aspx.cs:

```
using System;

namespace YourNamespace
{
    public partial class Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                // Check if text is stored in session and display it
                if (Session["StoredText"] != null)
                {
                    lblStoredText.Text = Session["StoredText"].ToString();
                }
            }
        }

        protected void btnSave_Click(object sender, EventArgs e)
        {
            // Store text in session
            Session["StoredText"] = txtInput.Text;
        }

        protected void btnRetrieve_Click(object sender, EventArgs e)
        {
            // Retrieve stored text from session and display it
            if (Session["StoredText"] != null)
            {
                lblStoredText.Text = Session["StoredText"].ToString();
            }
        }
    }
}
```

Q.2) Create an ASP.Net application, which show to the user 5-10 quiz questions. All Questions have 4 possible options and one right option exactly. Application counts and Shows to the user how many right answers were right and shows the result to user.

Solution:

QuizQuestions.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="QuizQuestions.aspx.cs"
Inherits="YourNamespace.QuizQuestions" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head runat="server">
   <title>Quiz Questions</title>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <h2>Quiz Questions</h2>
         <!-- Display quiz questions and options here -->
         <asp:Label ID="lblQuestion" runat="server"></asp:Label>
         <br />
         <asp:RadioButtonList ID="rbOptions" runat="server"></asp:RadioButtonList>
         <br />
         <asp:Button ID="btnNext" runat="server" Text="Next" OnClick="btnNext_Click" />
         <br />
         <asp:Label ID="lblResult" runat="server"></asp:Label>
      </div>
   </form>
</body>
</html>
```

QuizQuestions.aspx.cs:

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

namespace YourNamespace
{
   public partial class QuizQuestions : System.Web.UI.Page
   {
      private List<Question> quizQuestions;
      private int currentQuestionIndex;
      private int correctAnswersCount;

      protected void Page_Load(object sender, EventArgs e)
      {
         if (!IsPostBack)
         {
            // Initialize quiz questions and display the first question
            quizQuestions = GetQuizQuestions();
            currentQuestionIndex = 0;
            DisplayQuestion(currentQuestionIndex);
         }
      }

      private List<Question> GetQuizQuestions()
      {
         // Implement logic to fetch quiz questions from a data source or define them in code
         // Example:
         List<Question> questions = new List<Question>
         {
                new Question("Question 1: What is the capital of France?", new List<string>{"London",
"Paris", "Berlin", "Rome"}, 1),
                new Question("Question 2: Which planet is known as the Red Planet?", new
List<string>{"Venus", "Mars", "Jupiter", "Saturn"}, 1),
                // Add more questions here
            };
            return questions;
        }

        private void DisplayQuestion(int index)
        {
            lblQuestion.Text = quizQuestions[index].Text;
            rbOptions.DataSource = quizQuestions[index].Options;
            rbOptions.DataBind();
        }

        protected void btnNext_Click(object sender, EventArgs e)
        {
            // Evaluate user's response
            if (rbOptions.SelectedIndex != -1)
            {
                int selectedOptionIndex = rbOptions.SelectedIndex;
                if (selectedOptionIndex == quizQuestions[currentQuestionIndex].CorrectAnswerIndex)
                {
                    correctAnswersCount++;
                }

                // Move to the next question
                currentQuestionIndex++;
                if (currentQuestionIndex < quizQuestions.Count)
                {
                    DisplayQuestion(currentQuestionIndex);
                }
                else
                {
                    // Display quiz result
                    lblResult.Text = "Quiz completed. You answered " + correctAnswersCount + " out of " +
quizQuestions.Count + " questions correctly.";
                }
            }
            else
            {
                // Prompt the user to select an option
                lblResult.Text = "Please select an option.";
            }
        }
    }

    public class Question
    {
        public string Text { get; set; }
        public List<string> Options { get; set; }
        public int CorrectAnswerIndex { get; set; }

        public Question(string text, List<string> options, int correctAnswerIndex)
        {
            Text = text;
            Options = options;
            CorrectAnswerIndex = correctAnswerIndex;
        }
    }
}
```

Slip 21

Q. 1)Write a web application in asp.net to implement concept of state management using View State and query string.

Solution:

ViewStateQueryStringExample.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="ViewStateQueryStringExample.aspx.cs"
Inherits="YourNamespace.ViewStateQueryStringExample" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>ViewState and QueryString Example</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>ViewState and QueryString Example</h2>
            <asp:TextBox ID="txtInput" runat="server"></asp:TextBox>
            <br />
            <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />
            <br />
            <asp:Label ID="lblOutput" runat="server" Text=""></asp:Label>
        </div>
    </form>
</body>
</html>
```

ViewStateQueryStringExample.aspx.cs:

```
using System;

namespace YourNamespace
{
    public partial class ViewStateQueryStringExample : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                // Check if query string contains input value
                if (!string.IsNullOrEmpty(Request.QueryString["input"]))
                {
                    // Display input value from query string
                    string inputFromQueryString = Request.QueryString["input"];
                    lblOutput.Text = "Input from QueryString: " + inputFromQueryString;
                }
            }
        }

        protected void btnSubmit_Click(object sender, EventArgs e)
        {
            // Store input value in ViewState
            ViewState["input"] = txtInput.Text;

            // Redirect to the same page with input value in query string
            Response.Redirect("ViewStateQueryStringExample.aspx?input=" + txtInput.Text);
        }
    }
}
```

Q.2) Write an ASP.Net program to Display Employee details (EmpID, Name, Designation, Joining Date, Mob.no, Gender) from database Edit, Delete information from GridView

Solution:

EmployeeDetails.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="EmployeeDetails.aspx.cs"
Inherits="YourNamespace.EmployeeDetails" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Employee Details</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Employee Details</h2>
            <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
OnRowEditing="GridView1_RowEditing" OnRowDeleting="GridView1_RowDeleting">
                <Columns>
                    <asp:BoundField DataField="EmpID" HeaderText="EmpID" />
                    <asp:BoundField DataField="Name" HeaderText="Name" />
                    <asp:BoundField DataField="Designation" HeaderText="Designation" />
                    <asp:BoundField DataField="JoiningDate" HeaderText="Joining Date" />
                    <asp:BoundField DataField="MobNo" HeaderText="Mobile Number" />
                    <asp:BoundField DataField="Gender" HeaderText="Gender" />
                    <asp:CommandField ShowEditButton="True" />
                    <asp:CommandField ShowDeleteButton="True" />
                </Columns>
            </asp:GridView>
        </div>
    </form>
</body>
</html>
```

EmployeeDetails.aspx.cs:

```
using System;
using System.Data;
using System.Data.SqlClient;

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

        protected void BindGrid()
        {
            // Replace connection string and query with your actual database details
            string connectionString = "Your_Connection_String";
            string query = "SELECT EmpID, Name, Designation, JoiningDate, MobNo, Gender FROM
Employee";

            using (SqlConnection conn = new SqlConnection(connectionString))
            {
                SqlCommand cmd = new SqlCommand(query, conn);
                SqlDataAdapter da = new SqlDataAdapter(cmd);
                DataTable dt = new DataTable();

                conn.Open();
                da.Fill(dt);
                conn.Close();

                GridView1.DataSource = dt;
                GridView1.DataBind();
            }
        }
        protected void GridView1_RowEditing(object sender,
System.Web.UI.WebControls.GridViewEditEventArgs e)
        {
            // Set the grid view in edit mode
            GridView1.EditIndex = e.NewEditIndex;
            BindGrid();
        }

        protected void GridView1_RowDeleting(object sender,
System.Web.UI.WebControls.GridViewDeleteEventArgs e)
        {
            // Get the EmpID of the row being deleted
            int empID = Convert.ToInt32(GridView1.DataKeys[e.RowIndex].Values["EmpID"]);

            // Delete record from database
            string connectionString = "Your_Connection_String";
            string query = "DELETE FROM Employee WHERE EmpID = @EmpID";

            using (SqlConnection conn = new SqlConnection(connectionString))
            {
                SqlCommand cmd = new SqlCommand(query, conn);
                cmd.Parameters.AddWithValue("@EmpID", empID);

                conn.Open();
                cmd.ExecuteNonQuery();
                conn.Close();
            }

            // Rebind the grid view
            BindGrid();
        }
    }
}
```

Slip 22

Q.1) Show the use of Session State

Solution:

Default.aspx:

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

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Session State Example</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Session State Example</h2>
            <asp:TextBox ID="txtInput" runat="server"></asp:TextBox>
            <br />
            <asp:Button ID="btnSave" runat="server" Text="Save" OnClick="btnSave_Click" />
            <br />
            <asp:Label ID="lblStoredText" runat="server"></asp:Label>
        </div>
    </form>
</body>
</html>
```

Default.aspx.cs:

```
using System;

namespace YourNamespace
{
    public partial class Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                // Check if text is stored in session and display it
                if (Session["StoredText"] != null)
                {
                    lblStoredText.Text = Session["StoredText"].ToString();
                }
            }
        }

        protected void btnSave_Click(object sender, EventArgs e)
        {
            // Store text in session
            Session["StoredText"] = txtInput.Text;
            // Redirect to another page
            Response.Redirect("AnotherPage.aspx");
        }
    }
}
```

Q.2) Programs using ASP.NET Server controls. Create the application that accepts name, password ,age , email id, and user id. All the information entry is compulsory. Password should be reconfirmed. Age should be within 21 to 30. Email id should be valid. User id should have at least a capital letter and digit as well as length should be between 7 and 20 characters.

Solution:

RegistrationForm.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="RegistrationForm.aspx.cs"
Inherits="YourNamespace.RegistrationForm" %>

<!DOCTYPE html>
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Registration Form</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>User Registration</h2>
            <div>
                <asp:Label ID="lblName" runat="server" Text="Name: "></asp:Label>
                <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
                <asp:RequiredFieldValidator ID="rfvName" runat="server" ControlToValidate="txtName"
ErrorMessage="Name is required." Display="Dynamic" />
            </div>
            <br />
            <div>
                <asp:Label ID="lblPassword" runat="server" Text="Password: "></asp:Label>
                <asp:TextBox ID="txtPassword" runat="server" TextMode="Password"></asp:TextBox>
                <asp:RequiredFieldValidator ID="rfvPassword" runat="server"
ControlToValidate="txtPassword" ErrorMessage="Password is required." Display="Dynamic" />
            </div>
            <br />
            <div>
                <asp:Label ID="lblConfirmPassword" runat="server" Text="Confirm Password:
"></asp:Label>
                <asp:TextBox ID="txtConfirmPassword" runat="server"
TextMode="Password"></asp:TextBox>
                <asp:RequiredFieldValidator ID="rfvConfirmPassword" runat="server"
ControlToValidate="txtConfirmPassword" ErrorMessage="Confirm Password is required."
Display="Dynamic" />
                <asp:CompareValidator ID="cvPasswordMatch" runat="server"
ControlToValidate="txtConfirmPassword" ControlToCompare="txtPassword" Operator="Equal"
ErrorMessage="Passwords do not match." Display="Dynamic" />
            </div>
            <br />
            <div>
                <asp:Label ID="lblAge" runat="server" Text="Age: "></asp:Label>
                <asp:TextBox ID="txtAge" runat="server"></asp:TextBox>
                <asp:RequiredFieldValidator ID="rfvAge" runat="server" ControlToValidate="txtAge"
ErrorMessage="Age is required." Display="Dynamic" />
                <asp:RangeValidator ID="rvAge" runat="server" ControlToValidate="txtAge"
MinimumValue="21" MaximumValue="30" Type="Integer" ErrorMessage="Age must be between 21
and 30." Display="Dynamic" />
            </div>
            <br />
            <div>
                <asp:Label ID="lblEmail" runat="server" Text="Email ID: "></asp:Label>
                <asp:TextBox ID="txtEmail" runat="server"></asp:TextBox>
                <asp:RequiredFieldValidator ID="rfvEmail" runat="server" ControlToValidate="txtEmail"
ErrorMessage="Email ID is required." Display="Dynamic" />
```

```
        <asp:RegularExpressionValidator ID="revEmail" runat="server"
ControlToValidate="txtEmail" ErrorMessage="Invalid Email ID." ValidationExpression="\w+([-
+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*" Display="Dynamic" />
      </div>
      <br />
      <div>
        <asp:Label ID="lblUserID" runat="server" Text="User ID: "></asp:Label>
        <asp:TextBox ID="txtUserID" runat="server"></asp:TextBox>
        <asp:RequiredFieldValidator ID="rfvUserID" runat="server"
ControlToValidate="txtUserID" ErrorMessage="User ID is required." Display="Dynamic" />
        <asp:RegularExpressionValidator ID="revUserID" runat="server"
ControlToValidate="txtUserID" ErrorMessage="Invalid User ID." ValidationExpression="^(?=.*[A-
Z])(?=.*\d).{7,20}$" Display="Dynamic" />
      </div>
      <br />
      <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />
      <br />
      <asp:Label ID="lblMessage" runat="server" ForeColor="Green" Text=""></asp:Label>
    </div>
  </form>
</body>
</html>
```

RegistrationForm.aspx.cs:

```
using System;

namespace YourNamespace
{
  public partial class RegistrationForm : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {
      if (!IsPostBack)
      {
        // Clear any previous error messages
        lblMessage.Text = "";
      }
    }

    protected void btnSubmit_Click(object sender, EventArgs e)
    {
      if (Page.IsValid)
      {
        // Registration logic here
        // For demonstration, let's just display a success message
        lblMessage.Text = "Registration successful!";
      }
      else
      {
        lblMessage.Text = "Please correct the errors and try again.";
      }
    }
    }
}
```

Slip 23

Q.1) Show the use of Session State

Solution:

Default.aspx:

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

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Session State Example</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Session State Example</h2>
            <asp:TextBox ID="txtInput" runat="server"></asp:TextBox>
            <br />
            <asp:Button ID="btnSave" runat="server" Text="Save" OnClick="btnSave_Click" />
            <br />
            <asp:Label ID="lblStoredText" runat="server"></asp:Label>
        </div>
    </form>
</body>
</html>
```

Default.aspx.cs:

```
using System;

namespace YourNamespace
{
    public partial class Default : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                // Check if text is stored in session and display it
                if (Session["StoredText"] != null)
                {
                    lblStoredText.Text = Session["StoredText"].ToString();
                }
            }
        }

        protected void btnSave_Click(object sender, EventArgs e)
        {
            // Store text in session
            Session["StoredText"] = txtInput.Text;
            // Redirect to another page
            Response.Redirect("AnotherPage.aspx");
        }
    }
}
```

Q.2) Create the registration form using ASP.NET Server controls.
Solution:
RegistrationForm.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="RegistrationForm.aspx.cs"
Inherits="YourNamespace.RegistrationForm" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Registration Form</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>User Registration</h2>
            <div>
                <asp:Label ID="lblName" runat="server" Text="Name: "></asp:Label>
                <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
            </div>
            <br />
            <div>
                <asp:Label ID="lblPassword" runat="server" Text="Password: "></asp:Label>
                <asp:TextBox ID="txtPassword" runat="server" TextMode="Password"></asp:TextBox>
            </div>
            <br />
            <div>
                <asp:Label ID="lblConfirmPassword" runat="server" Text="Confirm Password:
"></asp:Label>
                <asp:TextBox ID="txtConfirmPassword" runat="server"
TextMode="Password"></asp:TextBox>
            </div>
            <br />
            <div>
                <asp:Label ID="lblAge" runat="server" Text="Age: "></asp:Label>
                <asp:TextBox ID="txtAge" runat="server"></asp:TextBox>
            </div>
            <br />
            <div>
                <asp:Label ID="lblEmail" runat="server" Text="Email ID: "></asp:Label>
                <asp:TextBox ID="txtEmail" runat="server"></asp:TextBox>
            </div>
            <br />
            <div>
                <asp:Label ID="lblUserID" runat="server" Text="User ID: "></asp:Label>
                <asp:TextBox ID="txtUserID" runat="server"></asp:TextBox>
            </div>
            <br />
            <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />
            <br />
            <asp:Label ID="lblMessage" runat="server" ForeColor="Green" Text=""></asp:Label>
        </div>
    </form>
</body>
</html>
```

RegistrationForm.aspx.cs:

```
using System;

namespace YourNamespace
{
    public partial class RegistrationForm : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                // Clear any previous error messages
                lblMessage.Text = "";
            }
        }

        protected void btnSubmit_Click(object sender, EventArgs e)
        {
            // Registration logic here
            // For demonstration, let's just display a success message
            lblMessage.Text = "Registration successful!";
        }
    }
}
```

Slip 24

Q.1) Write a Asp.Net program to display File upload and Calendar control.

Solution:

FileUploadAndCalendar.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="FileUploadAndCalendar.aspx.cs" Inherits="YourNamespace.FileUploadAndCalendar"
%>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title>File Upload and Calendar Control</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h2>File Upload</h2>
      <asp:FileUpload ID="FileUpload1" runat="server" /><br />
      <asp:Button ID="btnUpload" runat="server" Text="Upload" OnClick="btnUpload_Click" />
      <br /><br />
      <h2>Calendar Control</h2>
      <asp:Calendar ID="Calendar1" runat="server" />
    </div>
  </form>
</body>
</html>
```

FileUploadAndCalendar.aspx.cs:

```
using System;

namespace YourNamespace
{
  public partial class FileUploadAndCalendar : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {

    }

    protected void btnUpload_Click(object sender, EventArgs e)
    {
      if (FileUpload1.HasFile)
      {
        try
        {
          // Save the uploaded file to the server
          string fileName = System.IO.Path.GetFileName(FileUpload1.FileName);
          FileUpload1.SaveAs(Server.MapPath("~/uploads/") + fileName);
          // Display a success message
          Response.Write("File uploaded successfully!");
        }
        catch (Exception ex)
```

```
                {
                    // Display an error message if upload fails
                    Response.Write("Error: " + ex.Message);
                }
            }
            else
            {
                // Display a message if no file selected
                Response.Write("Please select a file to upload.");
            }
        }
    }
}
```

Q.2) Create the registration form using ASP.NET Server controls.

<u>Solution:</u>

RegistrationForm.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="RegistrationForm.aspx.cs"
Inherits="YourNamespace.RegistrationForm" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Registration Form</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>User Registration</h2>
            <div>
                <asp:Label ID="lblName" runat="server" Text="Name: "></asp:Label>
                <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
            </div>
            <br />
            <div>
                <asp:Label ID="lblEmail" runat="server" Text="Email: "></asp:Label>
                <asp:TextBox ID="txtEmail" runat="server"></asp:TextBox>
            </div>
            <br />
            <div>
                <asp:Label ID="lblPassword" runat="server" Text="Password: "></asp:Label>
                <asp:TextBox ID="txtPassword" runat="server" TextMode="Password"></asp:TextBox>
            </div>
            <br />
            <div>
                <asp:Label ID="lblConfirmPassword" runat="server" Text="Confirm Password:
"></asp:Label>
                <asp:TextBox ID="txtConfirmPassword" runat="server"
TextMode="Password"></asp:TextBox>
            </div>
            <br />
            <asp:Button ID="btnSubmit" runat="server" Text="Submit" OnClick="btnSubmit_Click" />
            <br />
            <asp:Label ID="lblMessage" runat="server" ForeColor="Green" Text=""></asp:Label>
        </div>
    </form>
</body>
</html>
```

RegistrationForm.aspx.cs:

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

namespace YourNamespace
{
    public partial class RegistrationForm : Page
    {
        protected void btnSubmit_Click(object sender, EventArgs e)
        {
            string name = txtName.Text;
            string email = txtEmail.Text;
            string password = txtPassword.Text;
            string confirmPassword = txtConfirmPassword.Text;

            if (password != confirmPassword)
            {
                lblMessage.Text = "Passwords do not match!";
            }
            else
            {
                // Your registration logic here
                // For demo purposes, just displaying a success message
                lblMessage.Text = "Registration successful!";
            }
        }
    }
}
```

Slip 25

Q.1) Write a Asp.Net program to display ---List of employees is available in listbox. Write an application to add selected or all records from listbox (assume multi-line property of textbox is true).

Solution:

ListBoxAndTextBox.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="ListBoxAndTextBox.aspx.cs"
Inherits="YourNamespace.ListBoxAndTextBox" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>ListBox and TextBox</title>
```

```
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h2>List of Employees</h2>
      <asp:ListBox ID="lstEmployees" runat="server" SelectionMode="Multiple">
        <asp:ListItem Text="John" />
        <asp:ListItem Text="Alice" />
        <asp:ListItem Text="Bob" />
        <asp:ListItem Text="Jane" />
        <asp:ListItem Text="Mike" />
      </asp:ListBox>
      <br />
      <h2>Selected Employees</h2>
      <asp:TextBox ID="txtSelectedEmployees" runat="server" TextMode="MultiLine"
Rows="5"></asp:TextBox>
      <br />
      <asp:Button ID="btnAddSelected" runat="server" Text="Add Selected"
OnClick="btnAddSelected_Click" />
      <asp:Button ID="btnAddAll" runat="server" Text="Add All" OnClick="btnAddAll_Click" />
    </div>
  </form>
</body>
</html>
```

ListBoxAndTextBox.aspx.cs:

```
using System;
using System.Linq;

namespace YourNamespace
{
  public partial class ListBoxAndTextBox : System.Web.UI.Page
  {
    protected void Page_Load(object sender, EventArgs e)
    {
      if (!IsPostBack)
      {
        // Populate the ListBox with employee names
        string[] employees = { "John", "Alice", "Bob", "Jane", "Mike" };
        lstEmployees.DataSource = employees;
        lstEmployees.DataBind();
      }
    }

    protected void btnAddSelected_Click(object sender, EventArgs e)
    {
      // Add selected items from ListBox to TextBox
      foreach (ListItem item in lstEmployees.Items)
      {
        if (item.Selected)
        {
                txtSelectedEmployees.Text += item.Text + Environment.NewLine;
            }
        }
    }

    protected void btnAddAll_Click(object sender, EventArgs e)
    {
        // Add all items from ListBox to TextBox
        foreach (ListItem item in lstEmployees.Items)
        {
            txtSelectedEmployees.Text += item.Text + Environment.NewLine;
        }
    }
}
}
```

Q.2) “How is the book ASP.NET with c# by Vipul Prakashan?” Give the user three choices: i) Good ii) Satisfactory iii) Bad. Provide a VOTE button. After user votes, present the result in percentage using labels next to the choices. (Use of radio buttons)

Solution:

VotingPage.aspx:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="VotingPage.aspx.cs"
Inherits="YourNamespace.VotingPage" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Voting Page</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>How is the book ASP.NET with C# by Vipul Prakashan?</h2>
            <asp:RadioButtonList ID="rblChoices" runat="server">
                <asp:ListItem Text="Good" Value="Good" />
                <asp:ListItem Text="Satisfactory" Value="Satisfactory" />
                <asp:ListItem Text="Bad" Value="Bad" />
            </asp:RadioButtonList>
            <br />
            <asp:Button ID="btnVote" runat="server" Text="Vote" OnClick="btnVote_Click" />
            <br />
            <asp:Label ID="lblGoodPercentage" runat="server" Text=""></asp:Label><br />
            <asp:Label ID="lblSatisfactoryPercentage" runat="server" Text=""></asp:Label><br />
            <asp:Label ID="lblBadPercentage" runat="server" Text=""></asp:Label>
        </div>
    </form>
</body>
</html>
```

VotingPage.aspx.cs:

```
using System;
using System.Linq;

namespace YourNamespace
{
   public partial class VotingPage : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {
         if (!IsPostBack)
         {
            // Initialize vote counts to zero
            Session["GoodCount"] = 0;
            Session["SatisfactoryCount"] = 0;
            Session["BadCount"] = 0;
         }
      }

      protected void btnVote_Click(object sender, EventArgs e)
      {
         // Update vote count based on user selection
         string selectedChoice = rblChoices.SelectedItem.Text;
         switch (selectedChoice)
         {
            case "Good":
               Session["GoodCount"] = (int)Session["GoodCount"] + 1;
               break;
            case "Satisfactory":
               Session["SatisfactoryCount"] = (int)Session["SatisfactoryCount"] + 1;
               break;
            case "Bad":
               Session["BadCount"] = (int)Session["BadCount"] + 1;
               break;
         }

         // Calculate and display percentages
         int totalVotes = (int)Session["GoodCount"] + (int)Session["SatisfactoryCount"] +
(int)Session["BadCount"];
         if (totalVotes > 0)
         {
            lblGoodPercentage.Text = "Good: " + ((double)(int)Session["GoodCount"] / totalVotes *
100).ToString("0.00") + "%";
            lblSatisfactoryPercentage.Text = "Satisfactory: " +
((double)(int)Session["SatisfactoryCount"] / totalVotes * 100).ToString("0.00") + "%";
            lblBadPercentage.Text = "Bad: " + ((double)(int)Session["BadCount"] / totalVotes *
100).ToString("0.00") + "%";
         }
      }
   }
}
```