L. J. INSTITUTE OF COMPUTER APPLICATION

Name: Bhandari Jaimini Nareshbhai

Roll No:03

Division: A

Enrollment No: 24002401110007

Subject: Web Application Development

1) **Create a web form for the following: Take two radio button. One for Gender and second for Hobbies. Take a button. On button click check whether both the radio button is selected or not. If not selected display the message "Please select Gender and Hobbies". If selected – show the selected options from Gender and Hobbies.**

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

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Gender and Hobbies Form</title>
</head>
<body>
   <form id="form1" runat="server">
      <div class="container">
         <h2>Select Gender and Hobby</h2>

         <h4>Gender:</h4>
         <asp:RadioButtonList ID="rblGender" runat="server">
            <asp:ListItem Text="Male" Value="Male"></asp:ListItem>
            <asp:ListItem Text="Female"
Value="Female"></asp:ListItem>
         </asp:RadioButtonList>

         <h4>Hobbies:</h4>
         <asp:RadioButtonList ID="rblHobby" runat="server">
            <asp:ListItem Text="Reading"
Value="Reading"></asp:ListItem>
            <asp:ListItem Text="Music" Value="Music"></asp:ListItem>
            <asp:ListItem Text="Sports" Value="Sports"></asp:ListItem>
         </asp:RadioButtonList>

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

```
            <div class="message">
                <asp:Label ID="lblMessage" runat="server"
Text=""></asp:Label>
            </div>
        </div>
    </form>
</body>
</html>

using System;
using System.Web.UI;

public partial class WebForm1 : Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void btnSubmit_Click(object sender, EventArgs e)
    {
        // Check if both radio button lists have a selected value
        if (string.IsNullOrEmpty(rblGender.SelectedValue) ||
string.IsNullOrEmpty(rblHobby.SelectedValue))
        {
            lblMessage.ForeColor = System.Drawing.Color.Red;
            lblMessage.Text = "Please select Gender and Hobbies.";
        }
        else
        {
            string gender = rblGender.SelectedValue;
            string hobby = rblHobby.SelectedValue;

            lblMessage.ForeColor = System.Drawing.Color.Green;
            lblMessage.Text = "Selected Gender: " + gender + "<br/>Selected
Hobby: " + hobby;}}}
```

2) **Create an application to take the following from the user: RollNo, Name, Subject1_marks, Subject2_marks, Subject3_marks, Subject4_marks, Subject5_marks. Take a button named "Calculate". On clicking calculate the total, percentage, and Pass/Fail. Add the validators.**

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

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Marks Calculator</title>
</head>
<body>
    <form id="form1" runat="server">
        <div class="container">
            <h2>Student Marks Calculator</h2>

            <div class="row">
                <label>Roll No:</label>
                <asp:TextBox ID="txtRollNo" runat="server"></asp:TextBox>
                <asp:RequiredFieldValidator ID="rfvRollNo" runat="server"
                    ControlToValidate="txtRollNo" ErrorMessage="* Required"
                    CssClass="error"></asp:RequiredFieldValidator>
            </div>

            <div class="row">
                <label>Name:</label>
                <asp:TextBox ID="txtName" runat="server"></asp:TextBox>
                <asp:RequiredFieldValidator ID="rfvName" runat="server"
                    ControlToValidate="txtName" ErrorMessage="* Required"
                    CssClass="error"></asp:RequiredFieldValidator>
            </div>

            <div class="row">
                <label>Subject 1 Marks:</label>
```

```
            <asp:TextBox ID="txtSub1" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="rfvSub1" runat="server"
               ControlToValidate="txtSub1" ErrorMessage="* Required"
CssClass="error"></asp:RequiredFieldValidator>
            <asp:RangeValidator ID="rvSub1" runat="server"
ControlToValidate="txtSub1"
               MinimumValue="0" MaximumValue="100" Type="Integer"
               ErrorMessage="* Enter 0–100"
CssClass="error"></asp:RangeValidator>
         </div>

         <div class="row">
            <label>Subject 2 Marks:</label>
            <asp:TextBox ID="txtSub2" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="rfvSub2" runat="server"
               ControlToValidate="txtSub2" ErrorMessage="* Required"
CssClass="error"></asp:RequiredFieldValidator>
            <asp:RangeValidator ID="rvSub2" runat="server"
ControlToValidate="txtSub2"
               MinimumValue="0" MaximumValue="100" Type="Integer"
               ErrorMessage="* Enter 0–100"
CssClass="error"></asp:RangeValidator>
         </div>

         <div class="row">
            <label>Subject 3 Marks:</label>
            <asp:TextBox ID="txtSub3" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="rfvSub3" runat="server"
               ControlToValidate="txtSub3" ErrorMessage="* Required"
CssClass="error"></asp:RequiredFieldValidator>
            <asp:RangeValidator ID="rvSub3" runat="server"
ControlToValidate="txtSub3"
               MinimumValue="0" MaximumValue="100" Type="Integer"
               ErrorMessage="* Enter 0–100"
CssClass="error"></asp:RangeValidator>
         </div>
```

```
        <div class="row">
            <label>Subject 4 Marks:</label>
            <asp:TextBox ID="txtSub4" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="rfvSub4" runat="server"
                ControlToValidate="txtSub4" ErrorMessage="* Required"
CssClass="error"></asp:RequiredFieldValidator>
            <asp:RangeValidator ID="rvSub4" runat="server"
ControlToValidate="txtSub4"
                MinimumValue="0" MaximumValue="100" Type="Integer"
                ErrorMessage="* Enter 0–100"
CssClass="error"></asp:RangeValidator>
        </div>

        <div class="row">
            <label>Subject 5 Marks:</label>
            <asp:TextBox ID="txtSub5" runat="server"></asp:TextBox>
            <asp:RequiredFieldValidator ID="rfvSub5" runat="server"
                ControlToValidate="txtSub5" ErrorMessage="* Required"
CssClass="error"></asp:RequiredFieldValidator>
            <asp:RangeValidator ID="rvSub5" runat="server"
ControlToValidate="txtSub5"
                MinimumValue="0" MaximumValue="100" Type="Integer"
                ErrorMessage="* Enter 0–100"
CssClass="error"></asp:RangeValidator>
        </div>

        <div class="row">
            <asp:Button ID="btnCalculate" runat="server"
Text="Calculate" OnClick="btnCalculate_Click" />
        </div>

        <div class="result">
            <asp:Label ID="lblResult" runat="server"
Text=""></asp:Label>
        </div>
      </div>
  </form>
```

```
</body>
</html>
using System;
using System.Web.UI;

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

    protected void btnCalculate_Click(object sender, EventArgs e)
    {
        // Run validation first
        Page.Validate();
        if (!Page.IsValid)
            return;

        // Get input values
        int sub1 = Convert.ToInt32(txtSub1.Text);
        int sub2 = Convert.ToInt32(txtSub2.Text);
        int sub3 = Convert.ToInt32(txtSub3.Text);
        int sub4 = Convert.ToInt32(txtSub4.Text);
        int sub5 = Convert.ToInt32(txtSub5.Text);

        int total = sub1 + sub2 + sub3 + sub4 + sub5;
        double percentage = total / 5.0;

        string result = (sub1 < 35 || sub2 < 35 || sub3 < 35 || sub4 < 35 || sub5
< 35)
                        ? "Fail"
                        : "Pass";

        lblResult.ForeColor = System.Drawing.Color.Blue;
        lblResult.Text = $"<br/>Roll No: {txtRollNo.Text}" +
                         $"<br/>Name: {txtName.Text}" +
                         $"<br/>Total Marks: {total}" +
```

```
                $"<br/>Percentage: {percentage:F2}%" +
                $"<br/>Result: {result}";
    }
}
```

3) **Write a C# program where a class has a constructor that accepts name and age. Use the constructor to initialize the object and display the result.**

```
using System;

namespace ConstructorExample
{
    class Person
    {
        // Data members
        private string name;
        private int age;

        // Constructor with parameters
        public Person(string name, int age)
        {
            this.name = name;
            this.age = age;
        }

        // Method to display details
        public void DisplayInfo()
        {
            Console.WriteLine("Name: " + name);
            Console.WriteLine("Age: " + age);
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            // Create object and initialize using constructor
            Person p1 = new Person("Harshil", 22);

            // Display result
            p1.DisplayInfo();
```

```
            Console.ReadLine();
        }
    }
}
```

4) **Create a Master Page having the following: • Header: "ABC University", with a logo image and navigation links (Home, About, Contact). • Footer: "© 2025 ABC University | Contact: abc@university.com". Create two content pages: • Page1: Accept student details (Roll No, Name, Course). Submit and display the data on Page2. • Page2: Display the submitted student details.**

Site.master :-

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeFile="Site.master.cs" Inherits="Site" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>LJ University</title>
</head>
<body>
   <form id="form1" runat="server">
      <header>
         <img src="~/images/logo.png" alt="University Logo" />
         <h1>ABC University</h1>
         <nav style="margin-left:auto;">
            <a href="Page1.aspx">Home</a>
            <a href="#">About</a>
            <a href="#">Contact</a>
         </nav>
      </header>

      <div class="content">
         <asp:ContentPlaceHolder ID="MainContent" runat="server" />
      </div>

      <footer>
         © 2025 ABC University | Contact: abc@university.com
      </footer>
   </form>
```

```
</body>
</html>
```

Page1.aspx :-

```
<%@ Page Title="Student Details" Language="C#"
MasterPageFile="~/Site.master" AutoEventWireup="true"
CodeFile="Page1.aspx.cs" Inherits="Page1" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent"
runat="server">
   <h2>Enter Student Details</h2>
   <table>
      <tr>
         <td>Roll No:</td>
         <td><asp:TextBox ID="txtRollNo"
runat="server"></asp:TextBox></td>
      </tr>
      <tr>
         <td>Name:</td>
         <td><asp:TextBox ID="txtName"
runat="server"></asp:TextBox></td>
      </tr>
      <tr>
         <td>Course:</td>
         <td><asp:TextBox ID="txtCourse"
runat="server"></asp:TextBox></td>
      </tr>
      <tr>
         <td colspan="2" style="text-align:center;">
            <asp:Button ID="btnSubmit" runat="server" Text="Submit"
OnClick="btnSubmit_Click" />
         </td>
      </tr>
   </table>
</asp:Content>
```

Page1.aspx.cs :-

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

public partial class Page1 : Page
{
    protected void btnSubmit_Click(object sender, EventArgs e)
    {
        // Store data in Session variables
        Session["RollNo"] = txtRollNo.Text;
        Session["Name"] = txtName.Text;
        Session["Course"] = txtCourse.Text;

        // Redirect to Page2
        Response.Redirect("Page2.aspx");
    }
}
```

Page2.aspx :-

```
<%@ Page Title="Student Details Display" Language="C#"
MasterPageFile="~/Site.master" AutoEventWireup="true"
CodeFile="Page2.aspx.cs" Inherits="Page2" %>

<asp:Content ID="Content1" ContentPlaceHolderID="MainContent"
runat="server">
    <h2>Student Details</h2>
    <asp:Label ID="lblDetails" runat="server" Text=""></asp:Label>
</asp:Content>
```

Page2.aspx.cs :-

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

public partial class Page2 : Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
```

```
        if (Session["RollNo"] != null && Session["Name"] != null &&
Session["Course"] != null)
        {
            lblDetails.Text = $"<b>Roll No:</b> {Session["RollNo"]}<br/>"
+
                              $"<b>Name:</b> {Session["Name"]}<br/>" +
                              $"<b>Course:</b> {Session["Course"]}";
        }
        else
        {
            lblDetails.Text = "No student data found. Please go back and enter
details.";
        }
    }
}
```

5) Create a web application which contains three Panels Panel: ProductSelection • Controls: A list of products (each with CheckBox or "Add" Button), Quantity selector per product, Button "View Cart" Panel: CartPanel • Controls: Grid/List of selected items (name, qty, unit price, subtotal), Label "Total", Button "Continue Shopping", Button "Checkout", Button "Remove Selected" Panel: CheckoutPanel (optional) • Controls: Billing/Shipping form (Name, Address, Payment method), Button "Back to Cart", Button "Place Order", Label for order confirmation Navigation Process: • Add (per product): Add the product and quantity to an in-memory cart; show confirmation message (e.g., "Added to cart"). • View Cart: Hide ProductSelection, show CartPanel populated with current cart items. • Continue Shopping: Hide CartPanel, show ProductSelection (cart persists). • Remove Selected: Delete selected items from the cart and refresh totals. • Checkout: Hide CartPanel, show CheckoutPanel. On successful Place Order, show confirmation and clear the cart.

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

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Simple Web Shop</title>
</head>
<body>
   <form id="form1" runat="server">
      <!-- Product Selection Panel -->
      <asp:Panel ID="PanelProductSelection" runat="server"
CssClass="panel">
         <h2>Select Products</h2>
         <table>
            <tr>
               <th>Product</th>
               <th>Price</th>
               <th>Quantity</th>
               <th></th>
```

```
            </tr>
            <tr>
                <td>Product A</td>
                <td>100</td>
                <td><asp:TextBox ID="txtQtyA" runat="server" Text="1"
Width="40px"></asp:TextBox></td>
                <td><asp:Button ID="btnAddA" runat="server" Text="Add"
OnClick="btnAddA_Click" /></td>
            </tr>
            <tr>
                <td>Product B</td>
                <td>200</td>
                <td><asp:TextBox ID="txtQtyB" runat="server" Text="1"
Width="40px"></asp:TextBox></td>
                <td><asp:Button ID="btnAddB" runat="server" Text="Add"
OnClick="btnAddB_Click" /></td>
            </tr>
            <tr>
                <td>Product C</td>
                <td>150</td>
                <td><asp:TextBox ID="txtQtyC" runat="server" Text="1"
Width="40px"></asp:TextBox></td>
                <td><asp:Button ID="btnAddC" runat="server" Text="Add"
OnClick="btnAddC_Click" /></td>
            </tr>
        </table>
        <br />
        <asp:Button ID="btnViewCart" runat="server" Text="View Cart"
CssClass="button" OnClick="btnViewCart_Click" />
        <br />
        <asp:Label ID="lblMessage" runat="server"
CssClass="message"></asp:Label>
    </asp:Panel>

    <!-- Cart Panel -->
    <asp:Panel ID="PanelCart" runat="server" CssClass="panel"
Visible="false">
```

```
        <h2>Your Cart</h2>
        <asp:GridView ID="gvCart" runat="server"
AutoGenerateColumns="False" Width="100%">
            <Columns>
                <asp:TemplateField HeaderText="Select">
                    <ItemTemplate>
                        <asp:CheckBox ID="chkSelect" runat="server" />
                    </ItemTemplate>
                </asp:TemplateField>
                <asp:BoundField DataField="ProductName"
HeaderText="Product" />
                <asp:BoundField DataField="Quantity" HeaderText="Qty"
/>
                <asp:BoundField DataField="UnitPrice" HeaderText="Unit
Price" />
                <asp:BoundField DataField="Subtotal"
HeaderText="Subtotal" />
            </Columns>
        </asp:GridView>

        <h3>Total: <asp:Label ID="lblTotal" runat="server"
Text="0"></asp:Label></h3>

        <asp:Button ID="btnContinueShopping" runat="server"
Text="Continue Shopping" OnClick="btnContinueShopping_Click" />
        <asp:Button ID="btnRemoveSelected" runat="server"
Text="Remove Selected" OnClick="btnRemoveSelected_Click" />
        <asp:Button ID="btnCheckout" runat="server" Text="Checkout"
OnClick="btnCheckout_Click" />
    </asp:Panel>

    <!-- Checkout Panel -->
    <asp:Panel ID="PanelCheckout" runat="server" CssClass="panel"
Visible="false">
        <h2>Checkout</h2>
        <table>
            <tr>
```

```
                <td>Name:</td>
                <td><asp:TextBox ID="txtName"
runat="server"></asp:TextBox></td>
            </tr>
            <tr>
                <td>Address:</td>
                <td><asp:TextBox ID="txtAddress" runat="server"
TextMode="MultiLine" Rows="3"></asp:TextBox></td>
            </tr>
            <tr>
                <td>Payment:</td>
                <td>
                    <asp:DropDownList ID="ddlPayment" runat="server">
                        <asp:ListItem>Credit Card</asp:ListItem>
                        <asp:ListItem>Debit Card</asp:ListItem>
                        <asp:ListItem>Cash on Delivery</asp:ListItem>
                    </asp:DropDownList>
                </td>
            </tr>
        </table>
        <br />
        <asp:Button ID="btnBackToCart" runat="server" Text="Back to
Cart" OnClick="btnBackToCart_Click" />
        <asp:Button ID="btnPlaceOrder" runat="server" Text="Place
Order" OnClick="btnPlaceOrder_Click" />
        <br /><br />
        <asp:Label ID="lblOrderConfirm" runat="server"
CssClass="message"></asp:Label>
      </asp:Panel>
  </form>
</body>
</html>

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

```
public partial class WebShop : Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            if (Session["Cart"] == null)
            {
                DataTable dt = new DataTable();
                dt.Columns.Add("ProductName");
                dt.Columns.Add("Quantity", typeof(int));
                dt.Columns.Add("UnitPrice", typeof(decimal));
                dt.Columns.Add("Subtotal", typeof(decimal));
                Session["Cart"] = dt;
            }
        }
    }

    // ---------- Helper: Add Product ----------
    private void AddToCart(string productName, decimal price, int qty)
    {
        DataTable dt = (DataTable)Session["Cart"];
        bool exists = false;

        foreach (DataRow row in dt.Rows)
        {
            if (row["ProductName"].ToString() == productName)
            {
                row["Quantity"] = Convert.ToInt32(row["Quantity"]) + qty;
                row["Subtotal"] = Convert.ToInt32(row["Quantity"]) * price;
                exists = true;
                break;
            }
        }

        if (!exists)
```

```
    {
        DataRow dr = dt.NewRow();
        dr["ProductName"] = productName;
        dr["Quantity"] = qty;
        dr["UnitPrice"] = price;
        dr["Subtotal"] = qty * price;
        dt.Rows.Add(dr);
    }

    Session["Cart"] = dt;
    lblMessage.Text = $"{productName} added to cart.";
}

protected void btnAddA_Click(object sender, EventArgs e)
{
    AddToCart("Product A", 100, int.Parse(txtQtyA.Text));
}

protected void btnAddB_Click(object sender, EventArgs e)
{
    AddToCart("Product B", 200, int.Parse(txtQtyB.Text));
}

protected void btnAddC_Click(object sender, EventArgs e)
{
    AddToCart("Product C", 150, int.Parse(txtQtyC.Text));
}

// ---------- View Cart ----------
protected void btnViewCart_Click(object sender, EventArgs e)
{
    ShowCartPanel();
}

private void ShowCartPanel()
{
    DataTable dt = (DataTable)Session["Cart"];
```

```
        gvCart.DataSource = dt;
        gvCart.DataBind();

        decimal total = 0;
        foreach (DataRow row in dt.Rows)
            total += Convert.ToDecimal(row["Subtotal"]);

        lblTotal.Text = total.ToString("F2");

        PanelProductSelection.Visible = false;
        PanelCart.Visible = true;
        PanelCheckout.Visible = false;
    }

    // ---------- Continue Shopping ----------
    protected void btnContinueShopping_Click(object sender, EventArgs
e)
    {
        PanelProductSelection.Visible = true;
        PanelCart.Visible = false;
        PanelCheckout.Visible = false;
    }

    // ---------- Remove Selected ----------
    protected void btnRemoveSelected_Click(object sender, EventArgs e)
    {
        DataTable dt = (DataTable)Session["Cart"];

        foreach (GridViewRow row in gvCart.Rows)
        {
            CheckBox chk = (CheckBox)row.FindControl("chkSelect");
            if (chk != null && chk.Checked)
            {
                string productName = row.Cells[1].Text;
                for (int i = dt.Rows.Count - 1; i >= 0; i--)
                {
                    if (dt.Rows[i]["ProductName"].ToString() == productName)
```

```
                {
                    dt.Rows.RemoveAt(i);
                    break;
                }
            }
        }
    }

    Session["Cart"] = dt;
    ShowCartPanel();
}

// ---------- Checkout ----------
protected void btnCheckout_Click(object sender, EventArgs e)
{
    PanelProductSelection.Visible = false;
    PanelCart.Visible = false;
    PanelCheckout.Visible = true;
}

// ---------- Back to Cart ----------
protected void btnBackToCart_Click(object sender, EventArgs e)
{
    PanelProductSelection.Visible = false;
    PanelCart.Visible = true;
    PanelCheckout.Visible = false;
}

// ---------- Place Order ----------
protected void btnPlaceOrder_Click(object sender, EventArgs e)
{
    // Normally you'd save to a database, but here we just simulate it
    lblOrderConfirm.Text = "Order placed successfully! Thank you, " +
txtName.Text + ".";
    Session["Cart"] = null; // clear cart
}
}
```

6) Create an application which has the following pages: (Apply Validations on all elements.) • Home page – with two options Login and New User • Login Page – If successfully logged in display the page to welcome the user • New User – Take Name, Contact No, Email and Birthdate for Registration. • After Successful login – Display a welcome message with all the details of registration.

Home.aspx

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

<!DOCTYPE html>
<html>
<head runat="server">
   <title>Home - User Portal</title>
</head>
<body>
   <form id="form1" runat="server">
      <h2>Welcome to User Portal</h2>
      <asp:Button ID="btnLogin" runat="server" Text="Login"
OnClick="btnLogin_Click" />
      <asp:Button ID="btnNewUser" runat="server" Text="New User"
OnClick="btnNewUser_Click" />
   </form>
</body>
</html>
```

Home.aspx.cs

```
using System;

public partial class Home : System.Web.UI.Page
{
   protected void btnLogin_Click(object sender, EventArgs e)
   {
      Response.Redirect("Login.aspx");
   }
```

```
    protected void btnNewUser_Click(object sender, EventArgs e)
    {
        Response.Redirect("NewUser.aspx");
    }
}
```

NewUser.aspx

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

<!DOCTYPE html>
<html>
<head runat="server">
    <title>New User Registration</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>New User Registration</h2>

        <table>
            <tr>
                <td>Name:</td>
                <td><asp:TextBox ID="txtName"
runat="server"></asp:TextBox></td>
                <td><asp:RequiredFieldValidator
ControlToValidate="txtName" runat="server" ErrorMessage="Enter
Name" ForeColor="Red" /></td>
            </tr>
            <tr>
                <td>Contact No:</td>
                <td><asp:TextBox ID="txtContact"
runat="server"></asp:TextBox></td>
                <td><asp:RequiredFieldValidator
ControlToValidate="txtContact" runat="server" ErrorMessage="Enter
Contact No" ForeColor="Red" /></td>
            </tr>
            <tr>
```

```
            <td>Email:</td>
            <td><asp:TextBox ID="txtEmail"
runat="server"></asp:TextBox></td>
            <td><asp:RequiredFieldValidator
ControlToValidate="txtEmail" runat="server" ErrorMessage="Enter
Email" ForeColor="Red" />
                <asp:RegularExpressionValidator
ControlToValidate="txtEmail" runat="server"
                    ValidationExpression="\w+@\w+\.\w+"
ErrorMessage="Invalid Email" ForeColor="Red" />
            </td>
        </tr>
        <tr>
            <td>Birthdate:</td>
            <td><asp:TextBox ID="txtBirthdate" runat="server"
TextMode="Date"></asp:TextBox></td>
            <td><asp:RequiredFieldValidator
ControlToValidate="txtBirthdate" runat="server" ErrorMessage="Enter
Birthdate" ForeColor="Red" /></td>
        </tr>
    </table>

    <br />
    <asp:Button ID="btnRegister" runat="server" Text="Register"
OnClick="btnRegister_Click" />
    <asp:Label ID="lblMessage" runat="server"
ForeColor="Green"></asp:Label>
  </form>
</body>
</html>
```

NewUser.aspx.cs

```
using System;

public partial class NewUser : System.Web.UI.Page
{
  protected void btnRegister_Click(object sender, EventArgs e)
```

```
    {
        // Save data in Session variables
        Session["Name"] = txtName.Text;
        Session["Contact"] = txtContact.Text;
        Session["Email"] = txtEmail.Text;
        Session["Birthdate"] = txtBirthdate.Text;

        lblMessage.Text = "Registration Successful! You can now Login.";
    }
}
```

Login.aspx

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

<!DOCTYPE html>
<html>
<head runat="server">
    <title>Login</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>User Login</h2>
        <table>
            <tr>
                <td>Email:</td>
                <td><asp:TextBox ID="txtEmail"
runat="server"></asp:TextBox></td>
                <td><asp:RequiredFieldValidator
ControlToValidate="txtEmail" runat="server" ErrorMessage="Enter
Email" ForeColor="Red" /></td>
            </tr>
            <tr>
                <td>Contact No:</td>
                <td><asp:TextBox ID="txtContact"
runat="server"></asp:TextBox></td>
```

```
            <td><asp:RequiredFieldValidator
ControlToValidate="txtContact" runat="server" ErrorMessage="Enter
Contact No" ForeColor="Red" /></td>
        </tr>
    </table>
    <br />
    <asp:Button ID="btnLogin" runat="server" Text="Login"
OnClick="btnLogin_Click" />
    <asp:Label ID="lblMessage" runat="server"
ForeColor="Red"></asp:Label>
  </form>
</body>
</html>
```

Login.aspx.cs

```
using System;

public partial class Login : System.Web.UI.Page
{
  protected void btnLogin_Click(object sender, EventArgs e)
  {
    // Check if session has registration data
    if (Session["Email"] != null && Session["Contact"] != null)
    {
      if (txtEmail.Text == Session["Email"].ToString() &&
txtContact.Text == Session["Contact"].ToString())
      {
        Response.Redirect("Welcome.aspx");
      }
      else
      {
        lblMessage.Text = "Invalid Email or Contact Number!";
      }
    }
    else
    {
      lblMessage.Text = "No user found! Please register first.";
```

```
        }
    }
}
```

Welcome.aspx

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

<!DOCTYPE html>
<html>
<head runat="server">
    <title>Welcome</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>Welcome Page</h2>
        <asp:Label ID="lblWelcome" runat="server" Font-
Bold="true"></asp:Label>
        <br /><br />
        <asp:Label ID="lblDetails" runat="server"></asp:Label>
    </form>
</body>
</html>
```

Welcome.aspx.cs

```
using System;

public partial class Welcome : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (Session["Name"] != null)
        {
            lblWelcome.Text = "Welcome, " + Session["Name"].ToString() +
"!";
            lblDetails.Text = "<br/>Contact: " + Session["Contact"] +
                        "<br/>Email: " + Session["Email"] +
```

```
                    "<br/>Birthdate: " + Session["Birthdate"];
        }
        else
        {
            Response.Redirect("Home.aspx");
        }
    }
}
```

7) Create three classes: • Person(Variables: Name, Basic) • Teacher(Variables:HRA-in % Method: TeacherSalary()) • Doctor(Variables:DA – in % Method: DoctorSalary ()). Teacher and Doctor Class inherits Person Class. A constructor in every class is used to initialize data members. Create Teacher and Doctor objects and use methods TeacherSalary() and DoctorSalary () to display the salary for the Teacher and Doctor respectively. Salary to be calculated as per the percentage of HRA and DA.

```
using System;

namespace InheritanceExample
{
   // Base class
   class Person
   {
      protected string Name;
      protected double Basic;

      // Constructor to initialize base class data
      public Person(string name, double basic)
      {
         Name = name;
         Basic = basic;
      }
   }

   // Derived class Teacher
   class Teacher : Person
   {
      private double HRA; // percentage

      // Constructor to initialize teacher data
      public Teacher(string name, double basic, double hra) : base(name,
basic)
      {
         HRA = hra;
```

```
        }

        // Method to calculate and display Teacher's salary
        public void TeacherSalary()
        {
            double salary = Basic + (Basic * HRA / 100);
            Console.WriteLine("----- Teacher Details -----");
            Console.WriteLine("Name: " + Name);
            Console.WriteLine("Basic Salary: " + Basic);
            Console.WriteLine("HRA (%): " + HRA);
            Console.WriteLine("Total Salary: " + salary);
            Console.WriteLine();
        }
    }

    // Derived class Doctor
    class Doctor : Person
    {
        private double DA; // percentage

        // Constructor to initialize doctor data
        public Doctor(string name, double basic, double da) : base(name,
basic)
        {
            DA = da;
        }

        // Method to calculate and display Doctor's salary
        public void DoctorSalary()
        {
            double salary = Basic + (Basic * DA / 100);
            Console.WriteLine("----- Doctor Details -----");
            Console.WriteLine("Name: " + Name);
            Console.WriteLine("Basic Salary: " + Basic);
            Console.WriteLine("DA (%): " + DA);
            Console.WriteLine("Total Salary: " + salary);
            Console.WriteLine();
```

```
    }
}

// Main class
class Program
{
    static void Main(string[] args)
    {
        // Creating Teacher object
        Teacher t = new Teacher("Harshil Patel", 30000, 20);
        t.TeacherSalary();

        // Creating Doctor object
        Doctor d = new Doctor("Dr. Mehta", 50000, 30);
        d.DoctorSalary();

        Console.ReadLine();
    }
}
}
```

8) Design a class to represent a bank account. Include the following members: Data members: • Name of the depositor • Account Number • Type of account • Balance amount in the account Methods: • To assign initial values • To deposit an amount • To withdraw an amount • To display the name and balance

```
using System;

namespace BankAccountExample
{
   class BankAccount
   {
      // Data members
      private string depositorName;
      private long accountNumber;
      private string accountType;
      private double balance;

      // Method to assign initial values
      public void AssignValues(string name, long accNo, string type,
double initialBalance)
      {
         depositorName = name;
         accountNumber = accNo;
         accountType = type;
         balance = initialBalance;
      }

      // Method to deposit amount
      public void Deposit(double amount)
      {
         if (amount > 0)
         {
            balance += amount;
            Console.WriteLine("Amount deposited successfully!");
         }
         else
```

```
        {
            Console.WriteLine("Deposit amount must be greater than
zero.");
        }
    }

    // Method to withdraw amount
    public void Withdraw(double amount)
    {
        if (amount > 0 && amount <= balance)
        {
            balance -= amount;
            Console.WriteLine("Amount withdrawn successfully!");
        }
        else if (amount > balance)
        {
            Console.WriteLine("Insufficient balance!");
        }
        else
        {
            Console.WriteLine("Invalid amount entered.");
        }
    }

    // Method to display name and balance
    public void Display()
    {
        Console.WriteLine("\n--- Account Details ---");
        Console.WriteLine("Depositor Name: " + depositorName);
        Console.WriteLine("Account Number: " + accountNumber);
        Console.WriteLine("Account Type: " + accountType);
        Console.WriteLine("Balance: " + balance);
    }
}

class Program
{
```

```
        static void Main(string[] args)
        {
            // Create an object of BankAccount
            BankAccount account = new BankAccount();

            // Assign initial values
            account.AssignValues("Harshil Patel", 1234567890, "Savings",
5000);

            // Display details
            account.Display();

            // Deposit amount
            Console.Write("\nEnter amount to deposit: ");
            double deposit = Convert.ToDouble(Console.ReadLine());
            account.Deposit(deposit);

            // Withdraw amount
            Console.Write("\nEnter amount to withdraw: ");
            double withdraw = Convert.ToDouble(Console.ReadLine());
            account.Withdraw(withdraw);

            // Display final details
            account.Display();

            Console.ReadLine();
        }
    }
}
```

9) Create an application for Employee Management (Create database in MySQL) Table: Employee(EmpId, Name, Department_Id, Salary) Page features: • GridView to show all employees. • Add buttons for the following: o Insert, Delete and Update operations. o Show the name of the highest paid employee. o Display the average salary o Show Department wise employee count

```
CREATE DATABASE EmployeeDB;

USE EmployeeDB;

CREATE TABLE Employee (
  EmpId INT PRIMARY KEY AUTO_INCREMENT,
  Name VARCHAR(50),
  Department_Id INT,
  Salary DECIMAL(10,2)
);

<configuration>
  <connectionStrings>
    <add name="MySqlConn"

connectionString="server=localhost;database=EmployeeDB;uid=root;pw
d=;"
         providerName="MySql.Data.MySqlClient" />
  </connectionStrings>
</configuration>

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

<!DOCTYPE html>
<html>
<head runat="server">
    <title>Employee Management</title>
</head>
<body>
    <form id="form1" runat="server">
        <h2>Employee Management System</h2>

        <asp:GridView ID="GridView1" runat="server"
AutoGenerateColumns="false"
            DataKeyNames="EmpId"
OnRowEditing="GridView1_RowEditing"
```

```
        OnRowCancelingEdit="GridView1_RowCancelingEdit"
        OnRowUpdating="GridView1_RowUpdating"
        OnRowDeleting="GridView1_RowDeleting"
        CellPadding="5" BorderColor="Black" BorderWidth="1">
        <Columns>
            <asp:BoundField DataField="EmpId" HeaderText="Emp ID"
ReadOnly="true" />
            <asp:BoundField DataField="Name" HeaderText="Name" />
            <asp:BoundField DataField="Department_Id"
HeaderText="Department ID" />
            <asp:BoundField DataField="Salary" HeaderText="Salary" />
            <asp:CommandField ShowEditButton="true"
ShowDeleteButton="true" />
        </Columns>
    </asp:GridView>

    <br />
    <h3>Add New Employee</h3>
    <table>
        <tr><td>Name:</td><td><asp:TextBox ID="txtName"
runat="server" /></td></tr>
        <tr><td>Department ID:</td><td><asp:TextBox ID="txtDeptId"
runat="server" /></td></tr>
        <tr><td>Salary:</td><td><asp:TextBox ID="txtSalary"
runat="server" /></td></tr>
    </table>
    <br />
    <asp:Button ID="btnInsert" runat="server" Text="Insert Employee"
OnClick="btnInsert_Click" />
    <hr />

    <asp:Button ID="btnHighestPaid" runat="server" Text="Show
Highest Paid Employee" OnClick="btnHighestPaid_Click" />
    <asp:Button ID="btnAverageSalary" runat="server" Text="Show
Average Salary" OnClick="btnAverageSalary_Click" />
    <asp:Button ID="btnDeptCount" runat="server" Text="Department
Wise Employee Count" OnClick="btnDeptCount_Click" />
```

```
        <br /><br />
        <asp:Label ID="lblResult" runat="server" Font-
Bold="true"></asp:Label>
    </form>
</body>
</html>

using System;
using System.Data;
using MySql.Data.MySqlClient;
using System.Configuration;

public partial class Employee : System.Web.UI.Page
{
    MySqlConnection con = new
MySqlConnection(ConfigurationManager.ConnectionStrings["MySqlCon
n"].ConnectionString);

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            LoadEmployees();
        }
    }

    void LoadEmployees()
    {
        MySqlDataAdapter da = new MySqlDataAdapter("SELECT *
FROM Employee", con);
        DataTable dt = new DataTable();
        da.Fill(dt);
        GridView1.DataSource = dt;
        GridView1.DataBind();
    }
```

```
    protected void btnInsert_Click(object sender, EventArgs e)
    {
        string query = "INSERT INTO Employee(Name, Department_Id,
Salary) VALUES (@Name, @Dept, @Salary)";
        MySqlCommand cmd = new MySqlCommand(query, con);
        cmd.Parameters.AddWithValue("@Name", txtName.Text);
        cmd.Parameters.AddWithValue("@Dept", txtDeptId.Text);
        cmd.Parameters.AddWithValue("@Salary", txtSalary.Text);

        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();

        lblResult.Text = "Employee Inserted Successfully!";
        LoadEmployees();
    }

    protected void GridView1_RowEditing(object sender,
System.Web.UI.WebControls.GridViewEditEventArgs e)
    {
        GridView1.EditIndex = e.NewEditIndex;
        LoadEmployees();
    }

    protected void GridView1_RowCancelingEdit(object sender,
System.Web.UI.WebControls.GridViewCancelEditEventArgs e)
    {
        GridView1.EditIndex = -1;
        LoadEmployees();
    }

    protected void GridView1_RowUpdating(object sender,
System.Web.UI.WebControls.GridViewUpdateEventArgs e)
    {
        int id = Convert.ToInt32(GridView1.DataKeys[e.RowIndex].Value);
```

```
        string name =
((System.Web.UI.WebControls.TextBox)GridView1.Rows[e.RowIndex].
Cells[1].Controls[0]).Text;
        string dept =
((System.Web.UI.WebControls.TextBox)GridView1.Rows[e.RowIndex].
Cells[2].Controls[0]).Text;
        string salary =
((System.Web.UI.WebControls.TextBox)GridView1.Rows[e.RowIndex].
Cells[3].Controls[0]).Text;

        string query = "UPDATE Employee SET Name=@Name,
Department_Id=@Dept, Salary=@Salary WHERE EmpId=@Id";
        MySqlCommand cmd = new MySqlCommand(query, con);
        cmd.Parameters.AddWithValue("@Name", name);
        cmd.Parameters.AddWithValue("@Dept", dept);
        cmd.Parameters.AddWithValue("@Salary", salary);
        cmd.Parameters.AddWithValue("@Id", id);

        con.Open();
        cmd.ExecuteNonQuery();
        con.Close();

        GridView1.EditIndex = -1;
        lblResult.Text = "Record Updated Successfully!";
        LoadEmployees();
    }

    protected void GridView1_RowDeleting(object sender,
System.Web.UI.WebControls.GridViewDeleteEventArgs e)
    {
        int id = Convert.ToInt32(GridView1.DataKeys[e.RowIndex].Value);

        string query = "DELETE FROM Employee WHERE EmpId=@Id";
        MySqlCommand cmd = new MySqlCommand(query, con);
        cmd.Parameters.AddWithValue("@Id", id);

        con.Open();
```

```
        cmd.ExecuteNonQuery();
        con.Close();

        lblResult.Text = "Record Deleted Successfully!";
        LoadEmployees();
    }

    protected void btnHighestPaid_Click(object sender, EventArgs e)
    {
        string query = "SELECT Name, Salary FROM Employee ORDER
BY Salary DESC LIMIT 1";
        MySqlCommand cmd = new MySqlCommand(query, con);

        con.Open();
        MySqlDataReader dr = cmd.ExecuteReader();
        if (dr.Read())
        {
            lblResult.Text = "Highest Paid Employee: " + dr["Name"] + "
(Salary: " + dr["Salary"] + ")";
        }
        dr.Close();
        con.Close();
    }

    protected void btnAverageSalary_Click(object sender, EventArgs e)
    {
        string query = "SELECT AVG(Salary) AS AvgSalary FROM
Employee";
        MySqlCommand cmd = new MySqlCommand(query, con);

        con.Open();
        object avg = cmd.ExecuteScalar();
        con.Close();

        lblResult.Text = "Average Salary: " + avg.ToString();
    }
```

```
    protected void btnDeptCount_Click(object sender, EventArgs e)
    {
        string query = "SELECT Department_Id, COUNT(*) AS
EmployeeCount FROM Employee GROUP BY Department_Id";
        MySqlDataAdapter da = new MySqlDataAdapter(query, con);
        DataTable dt = new DataTable();
        da.Fill(dt);

        lblResult.Text = "<b>Department Wise Employee
Count:</b><br/>";
        foreach (DataRow row in dt.Rows)
        {
            lblResult.Text += "Department " + row["Department_Id"] + ": " +
row["EmployeeCount"] + "<br/>";
        }
    }
}
```

10) Add data from the user in the list box. If the user enters same data give a message “Item already in the listbox”. Take a button named “Move” which add the selected items from first list box to another list box. (Do not delete from first listbox). Also show the selected items in a label.

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

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>ListBox Demo</title>
</head>
<body>
   <form id="form1" runat="server">
      <div class="container">
         <h2>ListBox Management Application</h2>

         <asp:TextBox ID="txtItem" runat="server" Placeholder="Enter
item"></asp:TextBox>
         <asp:Button ID="btnAdd" runat="server" Text="Add Item"
OnClick="btnAdd_Click" />
         <br /><br />

         <div class="listbox-container">
            <asp:ListBox ID="ListBox1" runat="server"
SelectionMode="Multiple" Height="150px"
Width="150px"></asp:ListBox>
            <asp:Button ID="btnMove" runat="server" Text="Move →"
OnClick="btnMove_Click" />
            <asp:ListBox ID="ListBox2" runat="server" Height="150px"
Width="150px"></asp:ListBox>
         </div>

         <br />
```

```
        <asp:Label ID="lblMessage" runat="server"
ForeColor="Red"></asp:Label><br />
        <asp:Label ID="lblSelected" runat="server"
ForeColor="Green"></asp:Label>
      </div>
   </form>
</body>
</html>

using System;
using System.Web.UI;

public partial class ListBoxDemo : Page
{
   protected void Page_Load(object sender, EventArgs e)
   {
      lblMessage.Text = "";
      lblSelected.Text = "";
   }

   protected void btnAdd_Click(object sender, EventArgs e)
   {
      string newItem = txtItem.Text.Trim();

      if (string.IsNullOrEmpty(newItem))
      {
         lblMessage.Text = "Please enter an item!";
         return;
      }

      // Check for duplicates
      bool exists = false;
      foreach (var item in ListBox1.Items)
      {
         if (item.ToString().Equals(newItem,
StringComparison.OrdinalIgnoreCase))
         {
```

```
            exists = true;
            break;
        }
    }

    if (exists)
    {
        lblMessage.Text = "Item already in the listbox!";
    }
    else
    {
        ListBox1.Items.Add(newItem);
        lblMessage.Text = "Item added successfully!";
    }

    txtItem.Text = "";
}

protected void btnMove_Click(object sender, EventArgs e)
{
    lblSelected.Text = "";

    if (ListBox1.GetSelectedIndices().Length == 0)
    {
        lblMessage.Text = "Please select at least one item to move.";
        return;
    }

    lblMessage.Text = "";

    foreach (var selectedItem in ListBox1.GetSelectedIndices())
    {
        string item = ListBox1.Items[(int)selectedItem].Text;
        ListBox2.Items.Add(item);
        lblSelected.Text += item + "<br/>";
    }
```

```
            lblMessage.Text = "Items moved successfully!";
        }
    }
```

11) Create the following using MVC: Take two classes: a. Student(RollNo, Name, ContactNo) b. Marks(RollNo, Sub1_Marks, Sub2_Marks, Sub3_Marks, Total, Percentage) Take input for RollNo, Name, ContactNo, Sub1_Marks, Sub2_Marks, Sub3_Marks. (Marks out of 100) Check that the student should secure minimum of 35 for passing in each subject. Calculate Total and Percentage. Display the Final result as Pass or Fail.

Models/Student.cs

```
using System.ComponentModel.DataAnnotations;

namespace MVCApp.Models
{
   public class Student
   {
      [Required]
      public int RollNo { get; set; }

      [Required]
      public string Name { get; set; }

      [Required]
      [Display(Name = "Contact No")]
      [RegularExpression(@"^[0-9]{10}$", ErrorMessage = "Enter valid
10-digit number")]
      public string ContactNo { get; set; }
   }
}
```

Models/Marks.cs

```
using System.ComponentModel.DataAnnotations;

namespace MVCApp.Models
{
   public class Marks
   {
      [Required]
```

```
        public int RollNo { get; set; }

        [Range(0, 100, ErrorMessage = "Marks must be between 0 and
100")]
        public int Sub1_Marks { get; set; }

        [Range(0, 100, ErrorMessage = "Marks must be between 0 and
100")]
        public int Sub2_Marks { get; set; }

        [Range(0, 100, ErrorMessage = "Marks must be between 0 and
100")]
        public int Sub3_Marks { get; set; }

        public int Total { get; set; }
        public double Percentage { get; set; }
        public string Result { get; set; }
    }
}
```

Controllers/StudentController.cs

```
using System.Web.Mvc;
using MVCApp.Models;

namespace MVCApp.Controllers
{
    public class StudentController : Controller
    {
        [HttpGet]
        public ActionResult Index()
        {
            return View();
        }

        [HttpPost]
        public ActionResult Index(Student student, Marks marks)
        {
```

```
            if (ModelState.IsValid)
            {
                // Calculate total and percentage
                marks.Total = marks.Sub1_Marks + marks.Sub2_Marks +
marks.Sub3_Marks;
                marks.Percentage = marks.Total / 3.0;

                // Check pass/fail condition
                if (marks.Sub1_Marks >= 35 && marks.Sub2_Marks >= 35
&& marks.Sub3_Marks >= 35)
                {
                    marks.Result = "Pass";
                }
                else
                {
                    marks.Result = "Fail";
                }

                // Pass data to Result view
                ViewBag.Student = student;
                ViewBag.Marks = marks;

                return View("Result");
            }

            return View();
        }
    }
}
```

Views/Student/Index.cshtml

```
@model MVCApp.Models.Student
@{
    ViewBag.Title = "Student Marks Entry";
}

<h2>Enter Student Details</h2>
```

```
@using (Html.BeginForm())
{
    <div>
        <label>Roll No:</label>
        @Html.TextBoxFor(m => m.RollNo, new { @class = "form-
control" })
        @Html.ValidationMessageFor(m => m.RollNo)
    </div>

    <div>
        <label>Name:</label>
        @Html.TextBoxFor(m => m.Name, new { @class = "form-control"
})
        @Html.ValidationMessageFor(m => m.Name)
    </div>

    <div>
        <label>Contact No:</label>
        @Html.TextBoxFor(m => m.ContactNo, new { @class = "form-
control" })
        @Html.ValidationMessageFor(m => m.ContactNo)
    </div>

    <hr />

    <h3>Enter Marks (out of 100)</h3>
    <div>
        <label>Subject 1:</label>
        <input type="number" name="Sub1_Marks" class="form-control"
min="0" max="100" required />
    </div>
    <div>
        <label>Subject 2:</label>
        <input type="number" name="Sub2_Marks" class="form-control"
min="0" max="100" required />
    </div>
```

```
    <div>
        <label>Subject 3:</label>
        <input type="number" name="Sub3_Marks" class="form-control"
min="0" max="100" required />
    </div>

    <br />
    <input type="submit" value="Calculate Result" class="btn btn-
primary" />
}

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

Views/Student/Result.cshtml

```
@{
    ViewBag.Title = "Result";
    var student = ViewBag.Student as MVCApp.Models.Student;
    var marks = ViewBag.Marks as MVCApp.Models.Marks;
}

<h2>Final Result</h2>

<table class="table table-bordered">
    <tr><th>Roll No</th><td>@student.RollNo</td></tr>
    <tr><th>Name</th><td>@student.Name</td></tr>
    <tr><th>Contact No</th><td>@student.ContactNo</td></tr>
    <tr><th>Subject 1 Marks</th><td>@marks.Sub1_Marks</td></tr>
    <tr><th>Subject 2 Marks</th><td>@marks.Sub2_Marks</td></tr>
    <tr><th>Subject 3 Marks</th><td>@marks.Sub3_Marks</td></tr>
    <tr><th>Total</th><td>@marks.Total</td></tr>

<tr><th>Percentage</th><td>@marks.Percentage.ToString("0.00")%</td
></tr>
    <tr><th>Result</th>
        <td style="color:@(marks.Result == "Pass" ? "green" : "red")">
```

```
            <b>@marks.Result</b>
        </td>
    </tr>
</table>

<a href="@Url.Action("Index", "Student")" class="btn btn-
success">Back</a>
```

12) Create the following using MVC: Classes: a. Patient(PatientId, Name, Age) b. Doctor(DoctorId, Name, Specialization) c. Report(PatientId, DoctorId, BP, Sugar, Result)
Input: Patient, Doctor, and health readings Process: • If BP > 140 or Sugar > 140 → "Abnormal" else "Normal" • Output: Patient Name, Doctor Name, and Diagnosis Result.

Models/Patient.cs

```
using System.ComponentModel.DataAnnotations;

namespace MVCApp.Models
{
   public class Patient
   {
      [Required]
      public int PatientId { get; set; }

      [Required]
      public string Name { get; set; }

      [Range(1, 120, ErrorMessage = "Enter a valid age")]
      public int Age { get; set; }
   }
}
```

Models/Doctor.cs

```
using System.ComponentModel.DataAnnotations;

namespace MVCApp.Models
{
   public class Doctor
   {
      [Required]
      public int DoctorId { get; set; }

      [Required]
      public string Name { get; set; }
```

```
        [Required]
        public string Specialization { get; set; }
    }
}
```

Models/Report.cs

```
using System.ComponentModel.DataAnnotations;

namespace MVCApp.Models
{
    public class Report
    {
        [Required]
        public int PatientId { get; set; }

        [Required]
        public int DoctorId { get; set; }

        [Range(0, 300, ErrorMessage = "BP must be between 0 and 300")]
        public double BP { get; set; }

        [Range(0, 300, ErrorMessage = "Sugar must be between 0 and
300")]
        public double Sugar { get; set; }

        public string Result { get; set; }
    }
}
```

Controllers/HealthController.cs

```
using System.Web.Mvc;
using MVCApp.Models;

namespace MVCApp.Controllers
{
    public class HealthController : Controller
```

```
    {
        [HttpGet]
        public ActionResult Index()
        {
            return View();
        }

        [HttpPost]
        public ActionResult Index(Patient patient, Doctor doctor, Report
report)
        {
            if (ModelState.IsValid)
            {
                // Diagnosis logic
                if (report.BP > 140 || report.Sugar > 140)
                    report.Result = "Abnormal";
                else
                    report.Result = "Normal";

                // Pass data to Result view
                ViewBag.Patient = patient;
                ViewBag.Doctor = doctor;
                ViewBag.Report = report;

                return View("Result");
            }

            return View();
        }
    }
}
```

Views/Health/Index.cshtml

```
@model MVCApp.Models.Patient
@{
    ViewBag.Title = "Health Diagnosis";
}
```

```
<h2>Patient Health Checkup</h2>

@using (Html.BeginForm())
{
    <fieldset>
        <legend>Patient Information</legend>
        <div>
            <label>Patient ID:</label>
            @Html.TextBoxFor(m => m.PatientId, new { @class = "form-
control" })
            @Html.ValidationMessageFor(m => m.PatientId)
        </div>
        <div>
            <label>Patient Name:</label>
            @Html.TextBoxFor(m => m.Name, new { @class = "form-
control" })
            @Html.ValidationMessageFor(m => m.Name)
        </div>
        <div>
            <label>Age:</label>
            @Html.TextBoxFor(m => m.Age, new { @class = "form-control"
})
            @Html.ValidationMessageFor(m => m.Age)
        </div>
    </fieldset>

    <fieldset>
        <legend>Doctor Information</legend>
        <div>
            <label>Doctor ID:</label>
            <input type="number" name="DoctorId" class="form-control"
required />
        </div>
        <div>
            <label>Doctor Name:</label>
```

```
            <input type="text" name="Name" class="form-control" required
/>
        </div>
        <div>
            <label>Specialization:</label>
            <input type="text" name="Specialization" class="form-control"
required />
        </div>
    </fieldset>

    <fieldset>
        <legend>Health Readings</legend>
        <div>
            <label>BP:</label>
            <input type="number" name="BP" class="form-control" required
/>
        </div>
        <div>
            <label>Sugar:</label>
            <input type="number" name="Sugar" class="form-control"
required />
        </div>
    </fieldset>

    <br />
    <input type="submit" value="Check Result" class="btn btn-primary"
/>
}

@section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
}
```

Views/Health/Result.cshtml

```
@{
    ViewBag.Title = "Diagnosis Result";
    var patient = ViewBag.Patient as MVCApp.Models.Patient;
```

```
    var doctor = ViewBag.Doctor as MVCApp.Models.Doctor;
    var report = ViewBag.Report as MVCApp.Models.Report;
}

<h2>Diagnosis Report</h2>

<table class="table table-bordered">
    <tr><th>Patient Name</th><td>@patient.Name</td></tr>
    <tr><th>Age</th><td>@patient.Age</td></tr>
    <tr><th>Doctor Name</th><td>@doctor.Name</td></tr>
    <tr><th>Specialization</th><td>@doctor.Specialization</td></tr>
    <tr><th>BP</th><td>@report.BP</td></tr>
    <tr><th>Sugar</th><td>@report.Sugar</td></tr>
    <tr>
        <th>Diagnosis Result</th>
        <td style="color:@(report.Result == "Normal" ? "green" : "red")">
            <b>@report.Result</b>
        </td>
    </tr>
</table>

<a href="@Url.Action("Index", "Health")" class="btn btn-
success">Back</a>
```