1. **Take three Button control Red, Green and Blue and one Label control. When a user presses any of the three buttons then the appearance of the label will change accordingly using CssClass.**

Aim : To implement the functionality of button clicked events.

Objective : To implement the functionality of button clicked events.

Theory :

- Button is an asp.net web server control. This control displays a push button control on the web page.
- button server control exists under **System.Web.UI.WebControls** namespace. button control allows the users to post a page to the web server. By default, a button control is a submit button.
- Button **OnClick()** method raises the click event of the button control.
- Button Click event occurs when the button control is clicked. the click event is commonly used when button control has no associated command name such as a submit button.
- Label is an asp.net web server control.This control display the text

Code:-

a. **Program.aspx:**

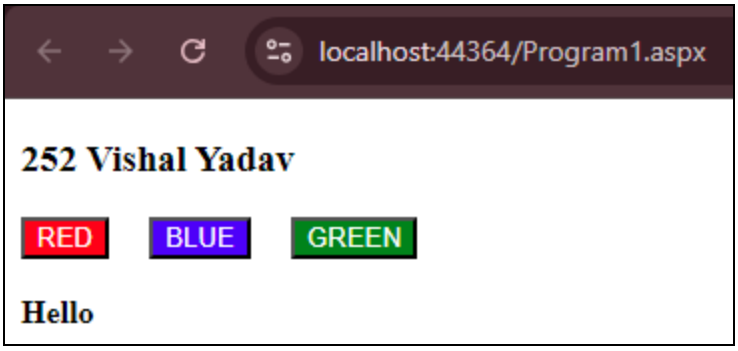
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="Program1.aspx.cs" Inherits="Vishal_252.Program1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
  <style type="text/css">
    .red{
      background-color : red;
      color:white;
    }
```

```
      .blue{
   background-color : blue;
   color:white;
         }

      .green{
   background-color : green;
    color:white;
     }
   </style>
</head>
<body>
   <form id="form1" runat="server">
      <div>
         <h3>252 Vishal Yadav</h3>


         <asp:Button ID="red" runat="server" Text="RED"
OnClick="red_Click" CssClass="red" />
   
         <asp:Button ID="blue" runat="server" Text="BLUE" CssClass="blue"
OnClick="blue_Click" />
   
         <asp:Button ID="green" runat="server" Text="GREEN"
CssClass="green" OnClick="green_Click" />
         <br />
         <br />
         <strong>
         <asp:Label ID="lbl_txt" runat="server" Text="Hello"></asp:Label>
         </strong>
         <br />


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

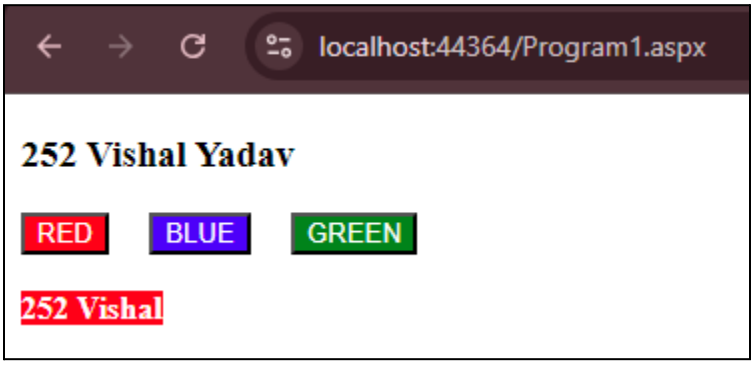
b. **Program.aspx.cs:**

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

namespace Vishal_252
{
   public partial class Program1 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {

      }

      protected void red_Click(object sender, EventArgs e)
      {
         lbl_txt.Text = "252 Vishal";
         lbl_txt.CssClass = "red";
      }

      protected void blue_Click(object sender, EventArgs e)
      {
         lbl_txt.Text = "252 Vishal";
         lbl_txt.CssClass = "blue";
      }

      protected void green_Click(object sender, EventArgs e)
      {
         lbl_txt.Text = "252 Vishal";
         lbl_txt.CssClass = "green";
      }
   }
}
```

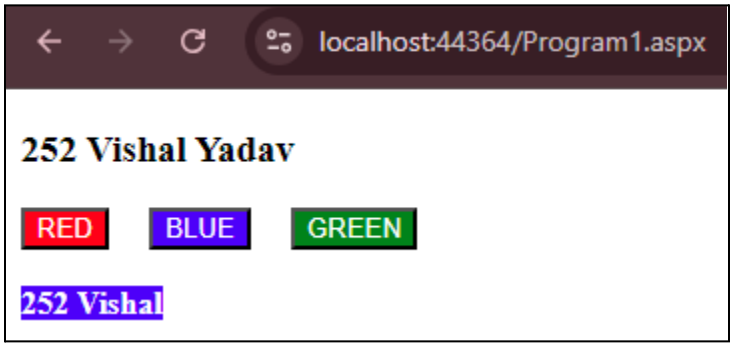
Output:-



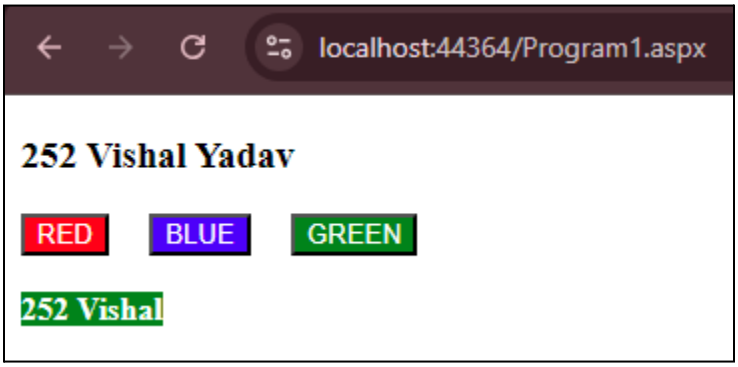
Red



Blue



Green



Conclusion: We have successfully implemented the functionality of button clicked events.

2. **Design a web application as follows. On submitting the form data confirm the selection made in the following format on one LABEL Control. Thank you very much _____. You have chosen _____ for breakfast. I will prepare it for you _____.**

Aim:
The aim of this web application is to collect user input regarding their name, juice preferences, and delivery time. Upon form submission, it displays a confirmation message summarizing their selections.

Objective:
To create a web application that collects user input for name, juice preferences, and delivery time, then displays a confirmation message summarizing their selections upon form submission.

Theory:
1. User Input: Collects name via a TextBox, juice preferences using CheckBoxes, and delivery time with RadioButtons.
2. Event Handling: Processes input on button click (`btn1_Click`).
3. Data Processing: Captures the name, concatenates selected juices, and determines delivery time.
4. Output Display: Shows a confirmation message in a Label summarizing the user's choices.

Code:-

Program2.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="Program2.aspx.cs" Inherits="Vishal_252.Program2" %>

<!DOCTYPE html>

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

```
<body>
  <form id="form1" runat="server">
    <div>
      <strong>252 Vishal Yadav</strong><br />
      <asp:Label ID="lb11" runat="server" Text="Please enter your name :
"></asp:Label>
      <asp:TextBox ID="tbx1" runat="server"></asp:TextBox>
      <br />
      <br />
    </div>
    <div>
      <asp:Label ID="lb2" runat="server" Text="What healthy juices you
want? "></asp:Label>
      <br />
      <br />
             
      <asp:CheckBox ID="chbx1" runat="server" Text="Apple" />
      <br />
             
      <asp:CheckBox ID="chbx2" runat="server" Text="Orange" />
      <br />
             
      <asp:CheckBox ID="chbx3" runat="server" Text="Carrot" />
      <br />
      <br />
    </div>

    <div>
      <asp:Label ID="lb13" runat="server" Text="Supply me
:"></asp:Label>
      <br />
      <br />
             
      <asp:RadioButton ID="rb1" runat="server" GroupName="Supply"
Text="Now" />
      <br />
             
      <asp:RadioButton ID="rb2" runat="server" GroupName="Supply"
Text="Later" />
      <br />
      <br />
    </div>
```

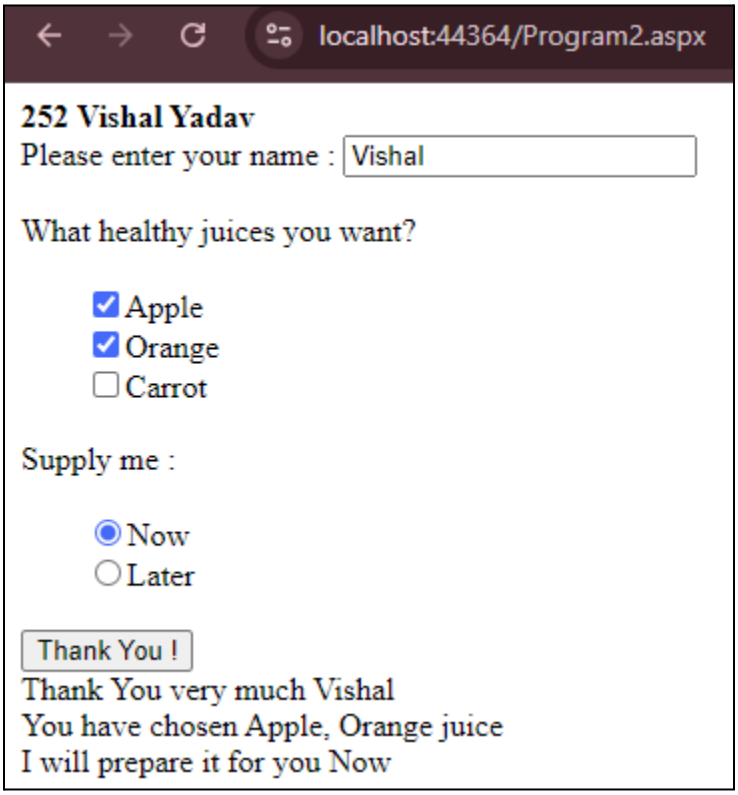
```
    <div>
      <asp:Button ID="btn1" runat="server" Text="Thank You !"
OnClick="btn1_Click" />
      <br />
      <asp:Label ID="lbl4" runat="server" Text=""></asp:Label>

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

Program2.aspx.cs

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

namespace Vishal_252
{
  public partial class Program2 : System.Web.UI.Page
  {
    public static string name, time, order;
    protected void Page_Load(object sender, EventArgs e)
    {
    }

    protected void btn1_Click(object sender, EventArgs e)
    {
      name = tbx1.Text;

      if (rb1.Checked)
      {
        time = rb1.Text;
      }
      if (rb2.Checked)
      {
        time = rb2.Text;
      }
```

```
        if (chbx1.Checked)
        {
           order = order + " " + chbx1.Text;
        }
        if (chbx2.Checked) {
           order = order + ", " + chbx2.Text;
        }
        if (chbx3.Checked)
        {
           order = order + ", " + chbx3.Text;
        }
        lbl4.Text = "Thank You very much "+name+"<br/>You have chosen
"+order+" juice "+"<br/>I will prepare it for you "+time;
 order = "";
      }
   }
}
```

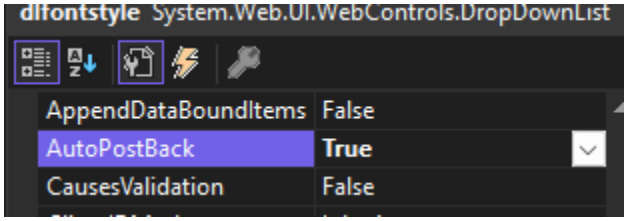
Output:-



Conclusion:
This web application effectively gathers user input using ASP.NET controls

and provides real-time feedback, enhancing user experience. It demonstrates basic form handling and event-driven programming.

3. **Write a program to give font effects (name, size, effect) to the text (without using Button control). Objective:To implement different font effects on a text**



Aim:
The aim of this program is to dynamically apply various font effects (name, size, style) to text in real-time without using a button control.

Objective:
To create a program that dynamically applies font effects (name, size, style) to text using dropdown selections, updating the display in real-time without button interaction.

Theory:
1. User Input : Utilizes DropDownList controls for font name, size, and style selection.
2. AutoPostBack : Enabled for DropDownList to automatically refresh the page when a selection is made.
3. Dynamic Display : Updates a Label with selected font attributes in real-time.
4. Event Handling : Uses SelectedIndexChanged events to apply changes immediately.

Code:-

Practical3.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="Practical3.aspx.cs" Inherits="Vishal_252.Practical3" %>

<!DOCTYPE html>
```

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

<strong>         &n
bsp;          &
nbsp;     252 Vishal Yadav<br />
      </strong>
      <br />
   

      <div>
                   
         <asp:DropDownList ID="dlfontstyle" runat="server"
AutoPostBack="True"
OnSelectedIndexChanged="dlfontstyle_SelectedIndexChanged">
   <asp:ListItem>Arial</asp:ListItem>
   <asp:ListItem>Times New Roman</asp:ListItem>
            <asp:ListItem>Verdana</asp:ListItem>
            <asp:ListItem>Brush Script MT</asp:ListItem>
</asp:DropDownList>


          &nbs
p;    

         <asp:DropDownList ID="dlfontsize" runat="server"
AutoPostBack="True"
OnSelectedIndexChanged="dlfontsize_SelectedIndexChanged">
            <asp:ListItem>1</asp:ListItem>
            <asp:ListItem>2</asp:ListItem>
            <asp:ListItem>3</asp:ListItem>
            <asp:ListItem>4</asp:ListItem>
            <asp:ListItem>5</asp:ListItem>
            <asp:ListItem>6</asp:ListItem>
            <asp:ListItem>7</asp:ListItem>
            <asp:ListItem>8</asp:ListItem>
            <asp:ListItem>9</asp:ListItem>
```

```
            <asp:ListItem>10</asp:ListItem>
            <asp:ListItem>11</asp:ListItem>
            <asp:ListItem>12</asp:ListItem>
            <asp:ListItem>13</asp:ListItem>
            <asp:ListItem>14</asp:ListItem>
            <asp:ListItem>15</asp:ListItem>
            <asp:ListItem>16</asp:ListItem>
            <asp:ListItem>17</asp:ListItem>
            <asp:ListItem>18</asp:ListItem>
            <asp:ListItem>19</asp:ListItem>
            <asp:ListItem>20</asp:ListItem>
</asp:DropDownList>
        <br />
        <br />
                  
        <asp:CheckBox ID="CheckBox1" runat="server" Text="Bold"
AutoPostBack="True" OnCheckedChanged="CheckBox1_CheckedChanged"
/>
   
        <asp:CheckBox ID="CheckBox2" runat="server" Text="Italic"
AutoPostBack="True" OnCheckedChanged="CheckBox2_CheckedChanged"
/>
   
        <asp:CheckBox ID="CheckBox3" runat="server" Text="Underline"
AutoPostBack="True" OnCheckedChanged="CheckBox3_CheckedChanged"
/>
        <br />
        <br />

          &nbs
p;         <asp:Label
ID="lb1" runat="server" Text="Hello, 252 Vishal"></asp:Label>
      </div>
   </form>
</body>
</html>
```

Practical3.aspx.cs

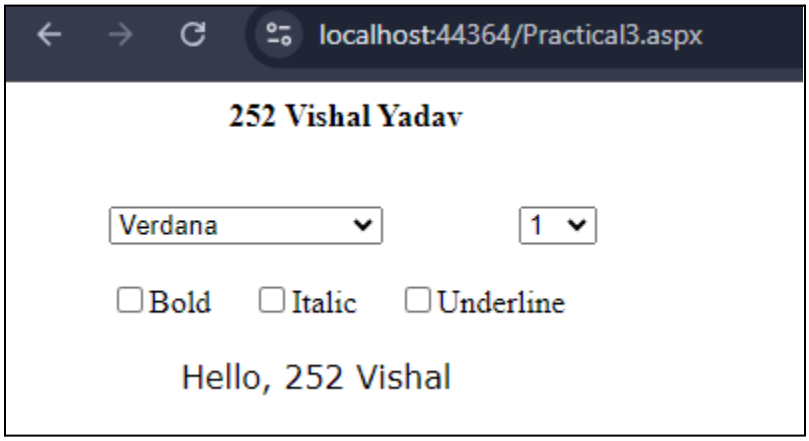
```
using System;
using System.Collections.Generic;
using System.Linq;
```

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

namespace Vishal_252
{
   public partial class Practical3 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {

      }

      protected void dlfontstyle_SelectedIndexChanged(object sender,
EventArgs e)
      {
         lb1.Font.Name = dlfontstyle.SelectedItem.Text.ToString();
      }

      protected void dlfontsize_SelectedIndexChanged(object sender,
EventArgs e)
      {
         lb1.Font.Size =
Convert.ToInt32(dlfontsize.SelectedItem.Text.ToString());
      }

      protected void CheckBox1_CheckedChanged(object sender, EventArgs e)
      {
         if (CheckBox1.Checked) {
            lb1.Font.Bold = true;
         }
         else
         {
            lb1.Font.Bold = false;
         }

      }

      protected void CheckBox2_CheckedChanged(object sender, EventArgs e)
      {
         if (CheckBox2.Checked) {
         lb1.Font.Italic = true;
```

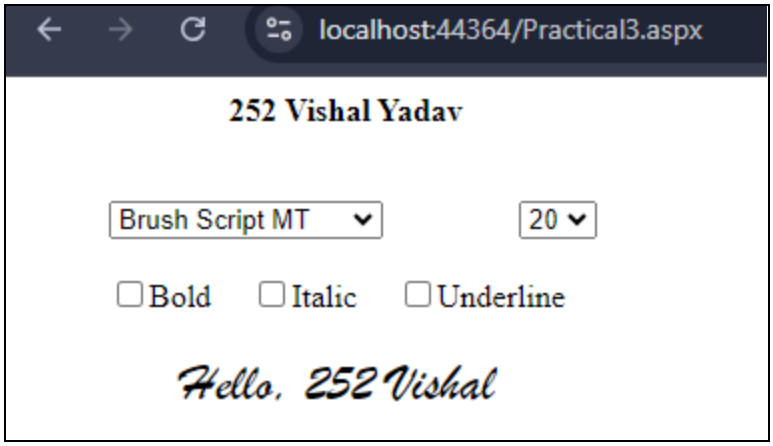
```
        }
        else
        {
          lb1.Font.Italic = false;
        }

      }

      protected void CheckBox3_CheckedChanged(object sender, EventArgs e)
      {
        if (CheckBox3.Checked) {
        lb1.Font.Underline = true;
        }
        else
        {
          lb1.Font.Underline = false;
        }
      }
    }
}
```

Output:-

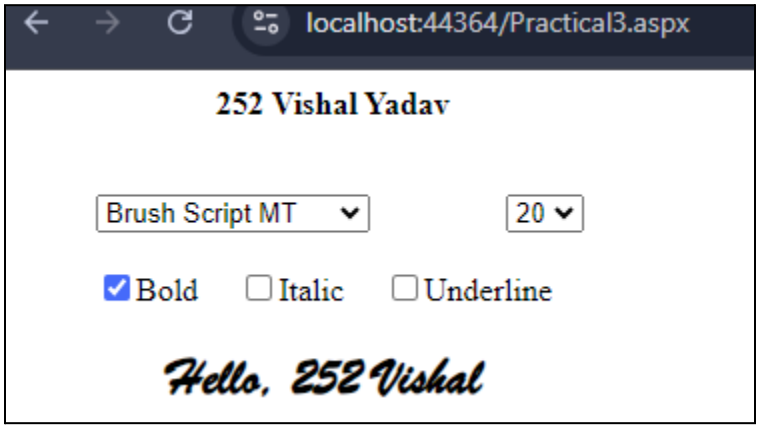
Font Style = Verdana



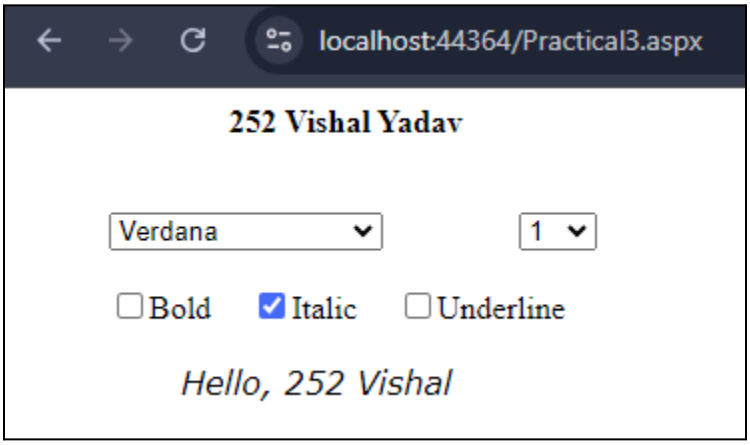
Font Size = 20



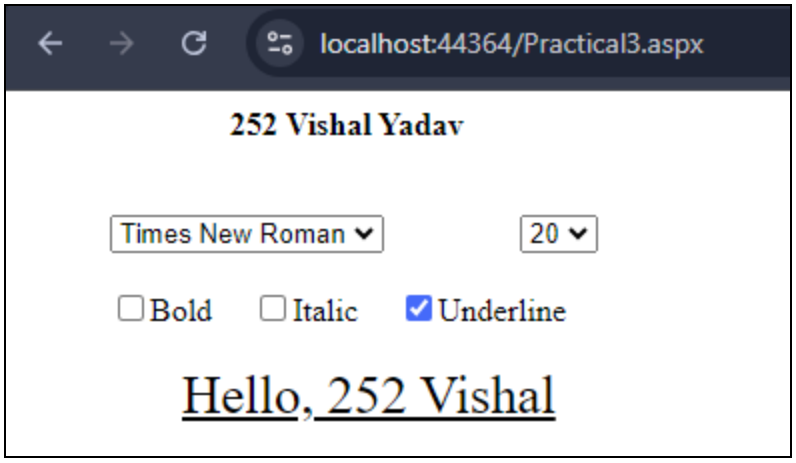
Bold



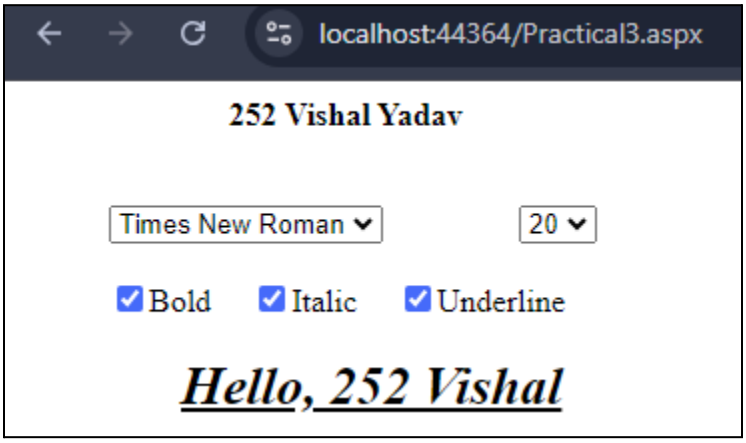
Italic



Underline



Checked Bold,Italic & Underline



Conclusion:
This program effectively demonstrates how to implement dynamic font effects in ASP.NET using AutoPostBack, enhancing user experience by allowing real-time customization of text appearance.

4. **Design an application that displays an image of a target (dart). If you click the center of the target, then a success message is displayed. Objective:To import an image ,display it and display a success message**

Aim:-
To create an interactive application that visually engages users by displaying a target image and providing feedback upon interaction.

Theory:-
The application utilizes event handling to detect clicks on specific regions of an image. By employing basic programming concepts such as image display, user interaction, and conditional logic, the app responds dynamically to user inputs.

Objective:-
To import and display a target image and provide a success message when the center of the target is clicked, demonstrating user interaction with graphical elements.

Code:-

WebForm1.aspx
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs"
Inherits="_252_Vishal_Yadav.WebForm1" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
    <div style="font-weight: 700">
      252 Vishal Yadav<br />
      <asp:ImageMap ID="ImageMap1" runat="server" Height="415px"
HotSpotMode="PostBack" ImageUrl="~/image/dart.jpg"
OnClick="ImageMap1_Click" Width="419px">
```

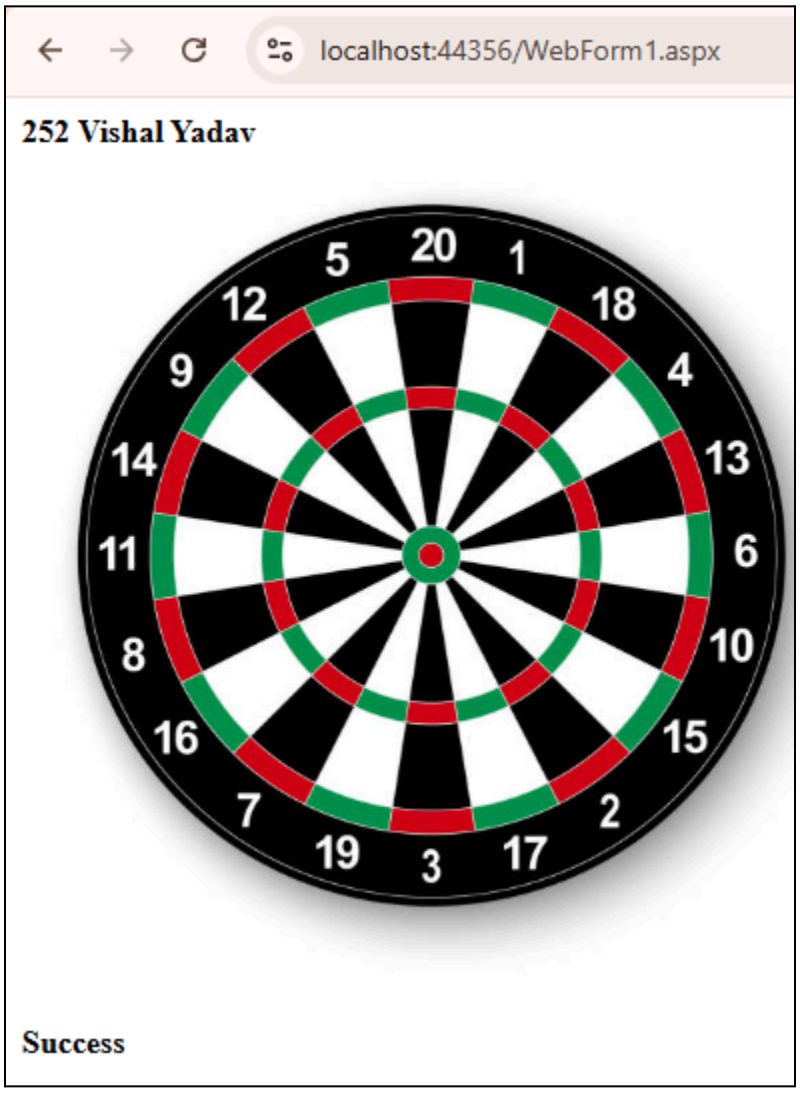
```
            <asp:CircleHotSpot HotSpotMode="PostBack"
NavigateUrl="~/WebForm2.aspx" PostBackValue="circle" Radius="50"
X="18" Y="205" />
            <asp:CircleHotSpot HotSpotMode="PostBack"
NavigateUrl="~/WebForm2.aspx" PostBackValue="circle" Radius="60"
X="211" Y="20" />
        </asp:ImageMap>
        <br />
        <br />
        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
        <br />
      </div>
   </form>
</body>
</html>
```

WebForm1.aspx.cs

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

namespace _252_Vishal_Yadav
{
   public partial class WebForm1 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {

      }

      protected void ImageMap1_Click(object sender, ImageMapEventArgs e)
      {
         if(e.PostBackValue == "circle")
         {
            Label1.Text = "Success";
         }
         else if(e.PostBackValue == "xyz")
         {
            Label1.Text = "Not Success";
```

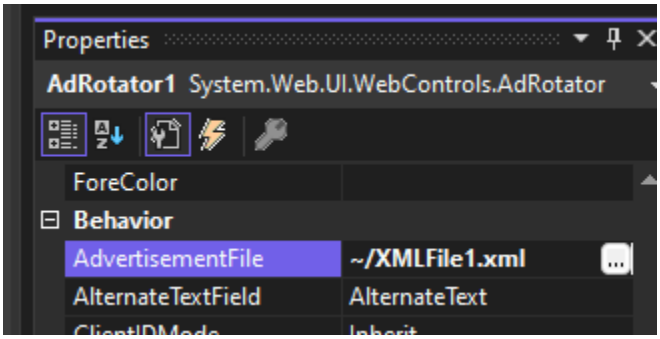
```
        }
      }
    }
}
```

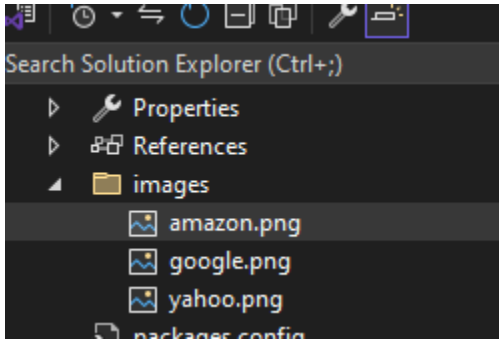
Output:-



Conclusion:-
This application illustrates fundamental programming concepts and enhances user engagement through visual feedback, showcasing the effectiveness of event-driven design in creating interactive experiences.

5. **Design an ASP.NET Application to Display Random Advertisements using ADRotator Control. Use XML DataSource and images from my Pictures.**





Aim:
The aim of this ASP.NET application is to display random advertisements using the **AdRotator** control, sourcing data from an XML file and displaying images from a specified directory.

Theory:
1. AdRotator Control: Utilizes the AdRotator control to display advertisements based on the XML data.
2. XML Data Source: The advertisement details, including image URLs and links, are stored in an XML file.
3. Image Directory: Images for the ads are stored in a designated folder (e.g., images).

Code:-

Webform1.aspx

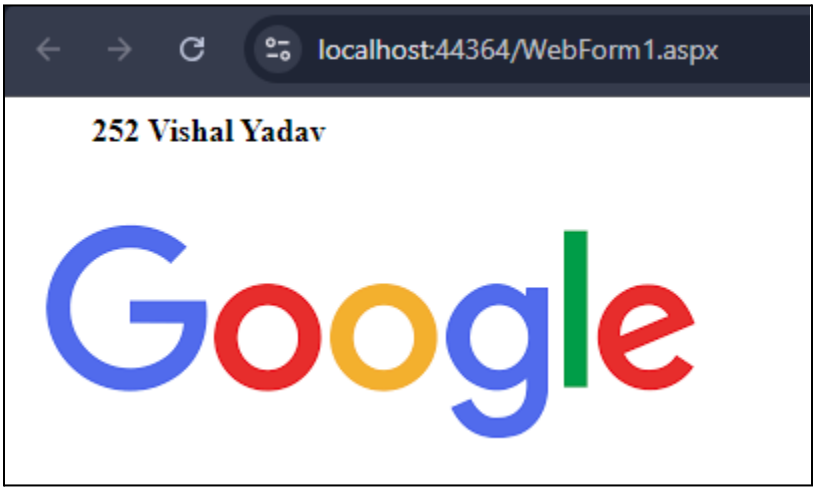
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="Vishal_252.WebForm1" %>

<!DOCTYPE html>

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

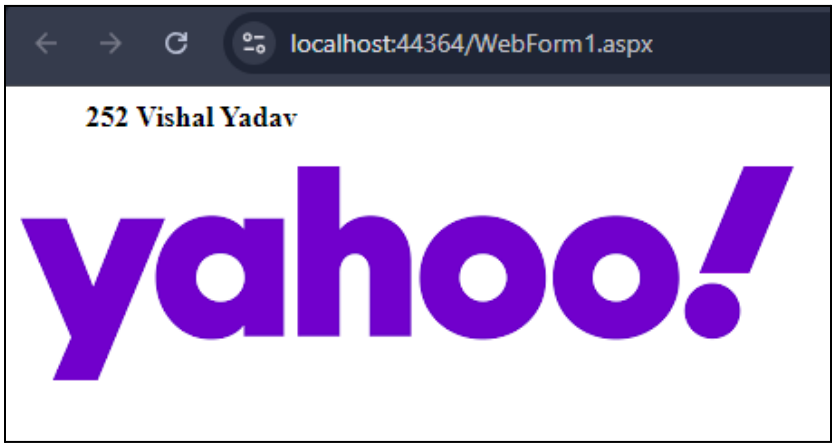
```
    <div>

                 <strong>252
Vishal Yadav</strong><br />
        <br />
        <asp:AdRotator ID="AdRotator1" runat="server"
AdvertisementFile="~/XMLFile1.xml" />

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

XmlFile1.xml

```
<?xml version="1.0" encoding="utf-8" ?>

<Advertisements>
        <Ad>
                <ImageUrl>images/google.png</ImageUrl>
                <NavigateUrl>https://google.com</NavigateUrl>
                <AlternateText>google </AlternateText>
                <Impressions>50</Impressions>
                <Keyword>google</Keyword>
        </Ad>
        <Ad>
                <ImageUrl>images/yahoo.png</ImageUrl>
                <NavigateUrl>http://yahoo.com</NavigateUrl>
                <AlternateText>yahoo</AlternateText>
                <Impressions>100</Impressions>
                <Keyword>yahoo</Keyword>
        </Ad>
        <Ad>
                <ImageUrl>images/amazon.png</ImageUrl>
                <NavigateUrl>https://amazon.com</NavigateUrl>
                <AlternateText>amazon</AlternateText>
                <Impressions>125</Impressions>
                <Keyword>amazon</Keyword>
        </Ad>
</Advertisements>
```

Output:-





Conclusion:
This application effectively demonstrates how to implement an advertisement rotation feature in ASP.NET using the AdRotator control and

XML data, enhancing user engagement with dynamic content.

6. **Write a program to upload your profile picture and display it in image control (file format should be .jpg, .jpeg or .png format)**

Aim:
To create a web application that allows users to upload their profile pictures and display them using image controls.

Objective:
To enable users to upload images in .jpg, .jpeg, or .png formats and display the uploaded images on the web page.

Theory:
1. File Upload: Uses the FileUpload control to select images.
2. File Validation: Checks file formats to ensure only valid image types are uploaded.
3. Image Display: Displays uploaded images using Image controls.
4. User Feedback: Provides messages indicating success or failure of the upload.

Code:-

Webform2.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm2.aspx.cs" Inherits="Vishal_252.WebForm2" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server" enctype="multipart/form-data">
    <div>

 <strong>        &n
bsp;   252 Vishal Yadav<br />
      <br />
```

```
        <br />
        <asp:FileUpload ID="FileUpload1" runat="server" />
        <br />
        <br />
        <asp:FileUpload ID="FileUpload2" runat="server" />
        <br />
        <br />
        <asp:Button ID="Button1" runat="server" Text="Upload" />
        <br />
        <br />
        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
        <br />
        <br />
        <asp:Image ID="Image1" runat="server" />
        <br />
        <br />
        <asp:Image ID="Image2" runat="server" />
        <br />
        </strong>
      </div>
   </form>
</body>
</html>
```

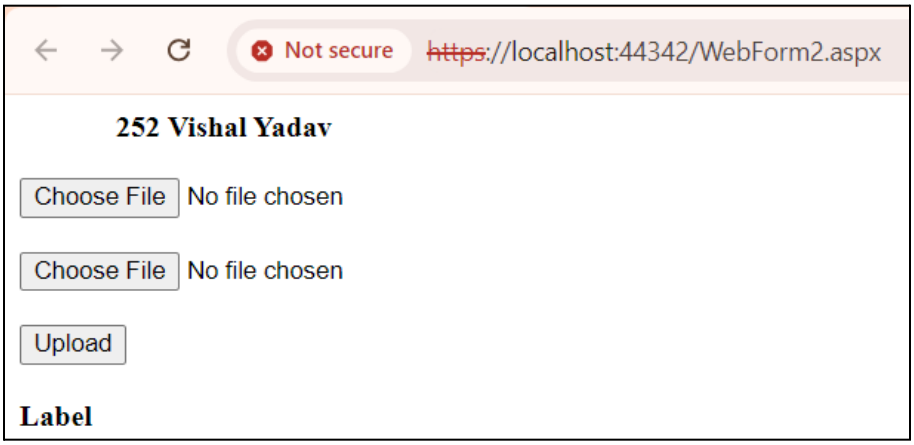
Webform2.aspx.cs

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

namespace Vishal_252
{
   public partial class WebForm2 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {

      }
```

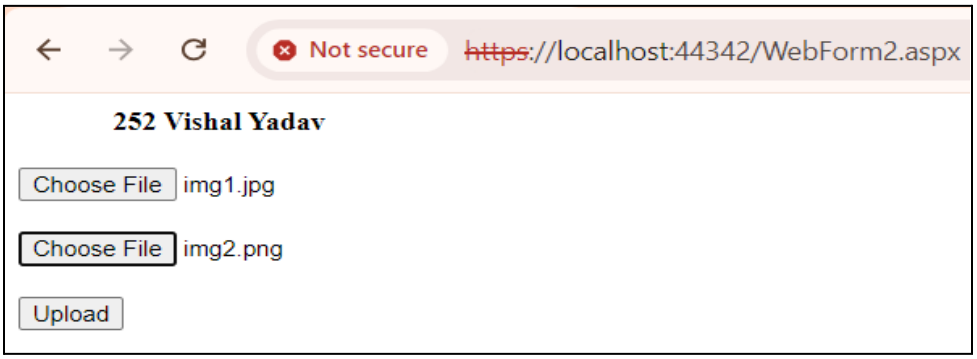
```
    protected void Button1_Click(object sender, EventArgs e)
    {
      if (FileUpload1.HasFile && FileUpload2.HasFile)
      {
        try
        {
          string fn = System.IO.Path.GetFileName(FileUpload1.FileName);
          string ext = System.IO.Path.GetExtension(fn);
          string fn2 = System.IO.Path.GetFileName(FileUpload2.FileName);
          string ext2 = System.IO.Path.GetExtension(fn2);
          if (ext == ".jpg")
            FileUpload1.SaveAs(Server.MapPath("Images\\") + fn);
          else if (ext2 == ".png")
            FileUpload2.SaveAs(Server.MapPath("Images\\") + fn2);
          Label1.ForeColor = System.Drawing.Color.Green;
          Label1.Text = "File Uploaded Successfully !!!";
          Image1.ImageUrl = "Images\\" + fn;
          Image1.Visible = true;
          Image2.ImageUrl = "Images\\" + fn2;
          Image2.Visible = true;
        }
        catch(Exception ex)
        {
          Label1.Text = "File could not be uploaded " + ex.Message;
        }
      }
      else
      {
        Image1.Visible = false;
        Image2.Visible = false;
        Label1.ForeColor = System.Drawing.Color.Red;
        Label1.Text = "No file Selected";
      }
    }
    }
  }
```

Output:-

Before



After



Conclusion:
This application successfully demonstrates image upload functionality, allowing users to upload and view their profile pictures while ensuring proper file format validation and providing user feedback.

7. **Design an ASP.NET application to Display Current Month's Calendar. Render the calendar to Display 1st May as Maharashtra Day.**

Aim:To design an application to display calendar

Objective:To implement calendar and display Maharashtra day

Theory:

- The calendar control has many properties and events, using which you can customize the actions and display of the control Object Sender is a parameter called Sender that contains a reference to the control/object that raised the event.

- Calendar.DayRender Event: Occurs when each day is created in the control hierarchy for the Calendar control.

Code:-

Webform3.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm3.aspx.cs" Inherits="Vishal_252.WebForm3" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
    <div style="padding-left:40px">
      <b>252 Vishal Yadav</b>
      <br />
      <div>
      <asp:Calendar ID="Calendar1" runat="server" BackColor="White"
BorderColor="Black" DayNameFormat="Shortest" Font-Names="Times
New Roman" Font-Size="10pt" ForeColor="Black" Height="220px"
NextPrevFormat="FullMonth" TitleFormat="Month" Width="400px"
OnDayRender="Calendar1_DayRender">
```

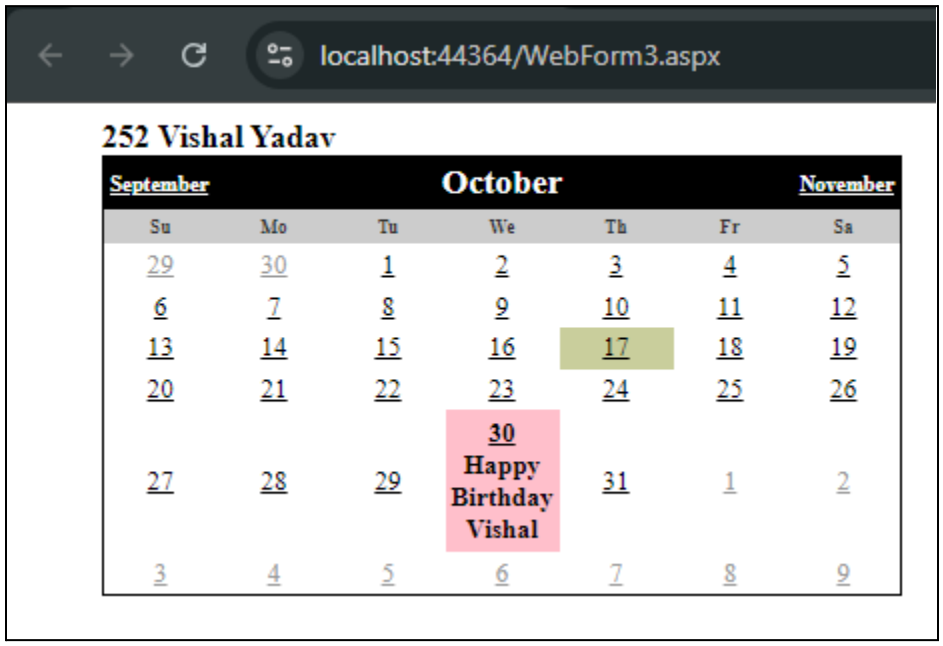
```
            <DayHeaderStyle BackColor="#CCCCCC" Font-Bold="True"
Font-Size="7pt" ForeColor="#333333" Height="10pt" />
            <DayStyle Width="14%" />
            <NextPrevStyle Font-Size="8pt" ForeColor="White" />
            <OtherMonthDayStyle ForeColor="#999999" />
            <SelectedDayStyle BackColor="#CC3333" ForeColor="White" />
            <SelectorStyle BackColor="#CCCCCC" Font-Bold="True"
Font-Names="Verdana" Font-Size="8pt" ForeColor="#333333"
Width="1%" />
            <TitleStyle BackColor="Black" Font-Bold="True" Font-Size="13pt"
ForeColor="White" Height="14pt" />
            <TodayDayStyle BackColor="#CCCC99" />
        </asp:Calendar>

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

Webform3.aspx.cs

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

namespace Vishal_252
{
   public partial class WebForm3 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {

      }

      protected void Calendar1_DayRender(object sender,
DayRenderEventArgs e)
      {
         DateTime d1 = new DateTime(2024, 10, 30);
```

```
        if (e.Day.Date.ToString() == d1.ToString())
        {
          e.Cell.Controls.Add(new LiteralControl("\nHappy Birthday Vishal"));
          e.Cell.BackColor = System.Drawing.Color.Pink; e.Cell.Font.Bold =
true;
          e.Cell.ForeColor = System.Drawing.Color.Black;
        }
      }
    }
}
```

Output:-



Conclusion:
The application effectively displays a calendar with my BirthDay highlighted, using the DayRender event for dynamic content and customization, enhancing user interaction.

8. **Write a program to demonstrate navigation controls in ASP.NET.**

Aim : To do the proper navigation flow.

Objective : To implement proper navigation controls to add flow to your web application.

Theory : Use of this control is very simple. You can add this control to your page then view your page in the browser. The Sitemap Path control displays the navigation path of the current page. The path acts as clickable links to previous pages.

Code:-

Web.SiteMap

```
<?xml version="1.0" encoding="utf-8" ?>
<siteMap xmlns="http://schemas.microsoft.com/AspNet/SiteMap-File-1.0"
>
  <siteMapNode url="Page1.aspx" title="Home"  description="">
    <siteMapNode url="Page2.aspx" title="About Us"  description="" />
    <siteMapNode url="Page3.aspx" title="Help"  description="" />
  </siteMapNode>

</siteMap>
```

Add in Web.config below <system.web>

```
 <siteMap>
  <providers>
          <remove name="MySqlSiteMapProvider"></remove>
  </providers>
 </siteMap>
```

Page1.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="Page1.aspx.cs" Inherits="Vishal_252.Page1" %>
```

```
<!DOCTYPE html>

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

         <asp:SiteMapPath ID="SiteMapPath1"
runat="server"></asp:SiteMapPath>
      </div>
   </form>
</body>
</html>
```

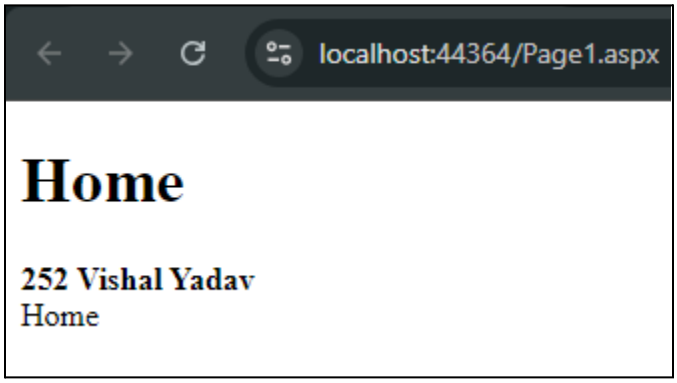
Page2.aspx

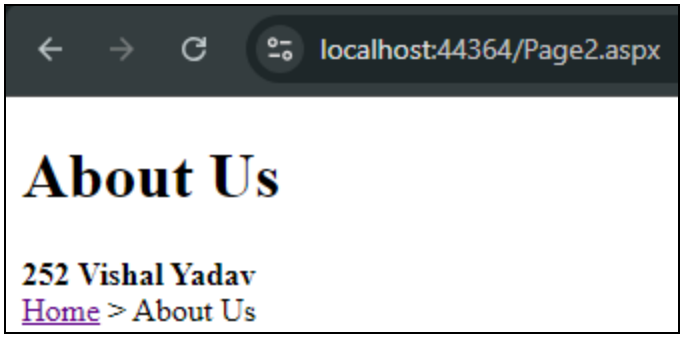
```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="Page2.aspx.cs" Inherits="Vishal_252.Page2" %>

<!DOCTYPE html>

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

         <asp:SiteMapPath ID="SiteMapPath1"
runat="server"></asp:SiteMapPath>
      </div>
   </form>
</body>
</html>
```

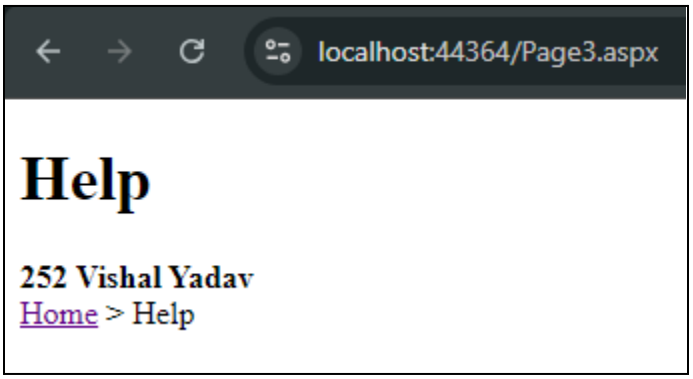
Page3.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="Page3.aspx.cs" Inherits="Vishal_252.Page3" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
    <div><h1>Help</h1></div>
    <div>
      <asp:SiteMapPath ID="SiteMapPath1"
runat="server"></asp:SiteMapPath>
    </div>
  </form>
</body>
</html>
```

Output:-







Conclusion:
This example demonstrates how to use ASP.NET's "SiteMapPath" control to create a simple navigation flow. It allows users to see and navigate through the site structure with clickable links. By defining a sitemap in `Web.sitemap` and integrating it into different pages, users can easily move between pages like Home, About Us, and Help.

9. **Create a registration form having following UI**

Aim:
Implement proper Validations by using all below Validation Controls:

a. All fields are required
b. Contact No must be 10 integer digits
c. Date must be selected with pop up date picker
d. Email must be in a valid form
e. Password must be 8 characters or greater and match with confirm password
f. At last display all the error messages

Objective:
To create a registration form that validates mandatory fields, contact numbers, email format, and password security, displaying any errors during submission.

Theory :

Validation

- Validation is an important part of any web application. User's input must always be validated before sending across different layers of the application.
- Validation controls are used to,
    - Implement presentation logic.
    - To validate user input data.
    - Data format, data type and data range is used for validation.
- There are six types of validation controls in ASP.NET
    1. RequiredFieldValidation Control

2. CompareValidator Control
3. RangeValidator Control
4. RegularExpressionValidator Control
5. CustomValidator Control
6. ValidationSummary

Validation Control	Description
RequiredFieldValidation	Makes an input control a required field
CompareValidator	Compares the value of one input control to the value of another input control or to a fixed value
RangeValidator	Checks that the user enters a value that falls between two values
RegularExpressionValidator	Ensures that the value of an input control matches a specified pattern
CustomValidator	Allows you to write a method to handle the validation of the value entered
ValidationSummary	Displays a report of all validation errors occurred in a Web page

Code:-

Web.Config

```
<appSettings>
                <add key="ValidationSettings:UnobtrusiveValidationMode"
value="None" />
</appSettings>
```

Webform3.aspx

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm3.aspx.cs" Inherits="Vishal_P1.WebForm3" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
  <title></title>
</head>
<body>
  <form id="form1" runat="server">
    <div style="padding-left:190px">
      <b>252_Vishal Yadav</b>
    </div>

     <asp:Panel ID="Panel1" runat="server" GroupingText="Registration
Form">
```

```
       
      <asp:Label ID="lb1" runat="server" Text="Enter Full
Name"></asp:Label>
                
 <asp:TextBox ID="txt_name" runat="server"
ValidationGroup="Registration"></asp:TextBox>
         
    <asp:RequiredFieldValidator ID="RequiredFieldValidator1"
runat="server"  ErrorMessage="Name cannot be empty please Enter."
ForeColor="Red" ControlToValidate="txt_name"
ValidationGroup="Registration"></asp:RequiredFieldValidator>
      <br />
        
      <br />
         
      <asp:Label ID="lb2" runat="server" Text="Address"></asp:Label>

          &nbs
p;          &nb
sp; 
    <asp:TextBox ID="txt_addr" runat="server" Rows="3"
ValidationGroup="Registration"></asp:TextBox>
       
    <asp:RequiredFieldValidator ID="RequiredFieldValidator2"
runat="server" ControlToValidate="txt_addr" Display="Dynamic"
ErrorMessage="Address cannot be empty please Enter." ForeColor="Red"
ValidationGroup="Registration"></asp:RequiredFieldValidator>

      <br />
        
      <br />
         
      <asp:Label ID="lb3" runat="server" Text="Select State"></asp:Label>

          &nbs
p;      
      <asp:DropDownList ID="dd_state" runat="server"
ValidationGroup="Registration" Width="160px">
        <asp:ListItem>Maharashtra</asp:ListItem>
        <asp:ListItem>Delhi</asp:ListItem>
        <asp:ListItem>Assam</asp:ListItem>
```

```
        </asp:DropDownList>
            
        <asp:RequiredFieldValidator ID="RequiredFieldValidator3"
runat="server" ControlToValidate="dd_state" ErrorMessage="Please Select
State" ForeColor="Red"
ValidationGroup="Registration"></asp:RequiredFieldValidator>
        <br />
          
        <br />
          
        <asp:Label ID="lb4" runat="server" Text="Mobile No."></asp:Label>

          &nbs
p;        
        <asp:TextBox ID="txt_contact" runat="server"
ValidationGroup="Registration"></asp:TextBox>
           
        <asp:RangeValidator ID="RangeValidator1" runat="server"
ControlToValidate="txt_contact" ErrorMessage="Please Enter Number is
Required" ForeColor="Red" MinimumValue="1000000000"
   MaximumValue="9999999999" Type="String"
ValidationGroup="Registration"></asp:RangeValidator>
        <br />
          
        <br />
          
        <asp:Label ID="lb5" runat="server" Text="Date Of
Birth"></asp:Label>

          &nbs
p;    
        <asp:TextBox ID="txt_dob" runat="server"
ValidationGroup="Registration"></asp:TextBox>
           
         <asp:RequiredFieldValidator ID="RequiredFieldValidator4"
runat="server" ControlToValidate="txt_dob" ErrorMessage="Please Enter
Date of Birth " ForeColor="Red"
ValidationGroup="Registration"></asp:RequiredFieldValidator>
      <asp:Calendar ID="Calendar1" runat="server" Height="16px"
Width="135px"
OnSelectionChanged="Calendar1_SelectionChanged"></asp:Calendar>
        <br />
```

```
<br />
          
        <asp:Label ID="lb6" runat="server" Text="Email"></asp:Label>

          &nbs
p;          &nb
sp;    <asp:TextBox ID="txt_email"
runat="server" ValidationGroup="Registration"></asp:TextBox>
            
        <asp:RequiredFieldValidator ID="RequiredFieldValidator7"
runat="server" ControlToValidate="txt_email" ErrorMessage="Please
Enter Email" ForeColor="Red"
ValidationGroup="Registration"></asp:RequiredFieldValidator>

         <asp:RegularExpressionValidator ID="RegularExpressionValidator1"
runat="server" ControlToValidate="txt_email" ErrorMessage="Please
Enter Valid Email Id" ForeColor="Red"
ValidationExpression="\w+([-+.']\w+)*@\w+([-.]\w+)*\.\w+([-.]\w+)*"
ValidationGroup="Registration"></asp:RegularExpressionValidator>
  <br />
          
        <br />
          
        <asp:Label ID="lb7" runat="server" Text="Password"></asp:Label>

          &nbs
p;        
        <asp:TextBox ID="txt_pwd" runat="server"
ValidationGroup="Registration"></asp:TextBox>
           
         <asp:RequiredFieldValidator ID="RequiredFieldValidator5"
runat="server" ControlToValidate="txt_pwd" ErrorMessage="Please Enter
Password" ForeColor="Red"
ValidationGroup="Registration"></asp:RequiredFieldValidator>

        <asp:CustomValidator ID="CustomValidator1" runat="server"
ControlToValidate="txt_pwd" ErrorMessage="Please Enter Valid Password"
ForeColor="Red" OnServerValidate="CustomValidator1_ServerValidate"
ValidationGroup="Registration"></asp:CustomValidator>
        <br />
          
        <br />
```

```
          
        <asp:Label ID="lb8" runat="server" Text="Confirm
Password"></asp:Label>
             
        <asp:TextBox ID="txt_pwd_confirm" runat="server"
ValidationGroup="Registration"></asp:TextBox>
           
        <asp:RequiredFieldValidator ID="RequiredFieldValidator6"
runat="server" ControlToValidate="txt_pwd_confirm"
ErrorMessage="Please Confirm Password" ForeColor="Red"
ValidationGroup="Registration"></asp:RequiredFieldValidator>

       <asp:CompareValidator ID="CompareValidator1" runat="server"
ControlToCompare="txt_pwd_confirm" ControlToValidate="txt_pwd"
ErrorMessage="Invalid Password" ForeColor="Red"
ValidationGroup="Registration"></asp:CompareValidator>

       <br />
      <br />

             

       <asp:Button ID="btn_submit" runat="server" Height="31px"
OnClick="btn_submit_Click" Text="Register"
ValidationGroup="Registration" Width="70px" />

          &nbs
p;          &nb
sp;         
        <asp:Button ID="btn_cancel" runat="server" Height="31px"
OnClick="btn_cancel_Click" Text="Cancel"
ValidationGroup="Registration" Width="66px" />
       <br />
        <br />
        <asp:Label ID="Label" runat="server"></asp:Label>
        <br />

      </asp:Panel>



   </form>
```

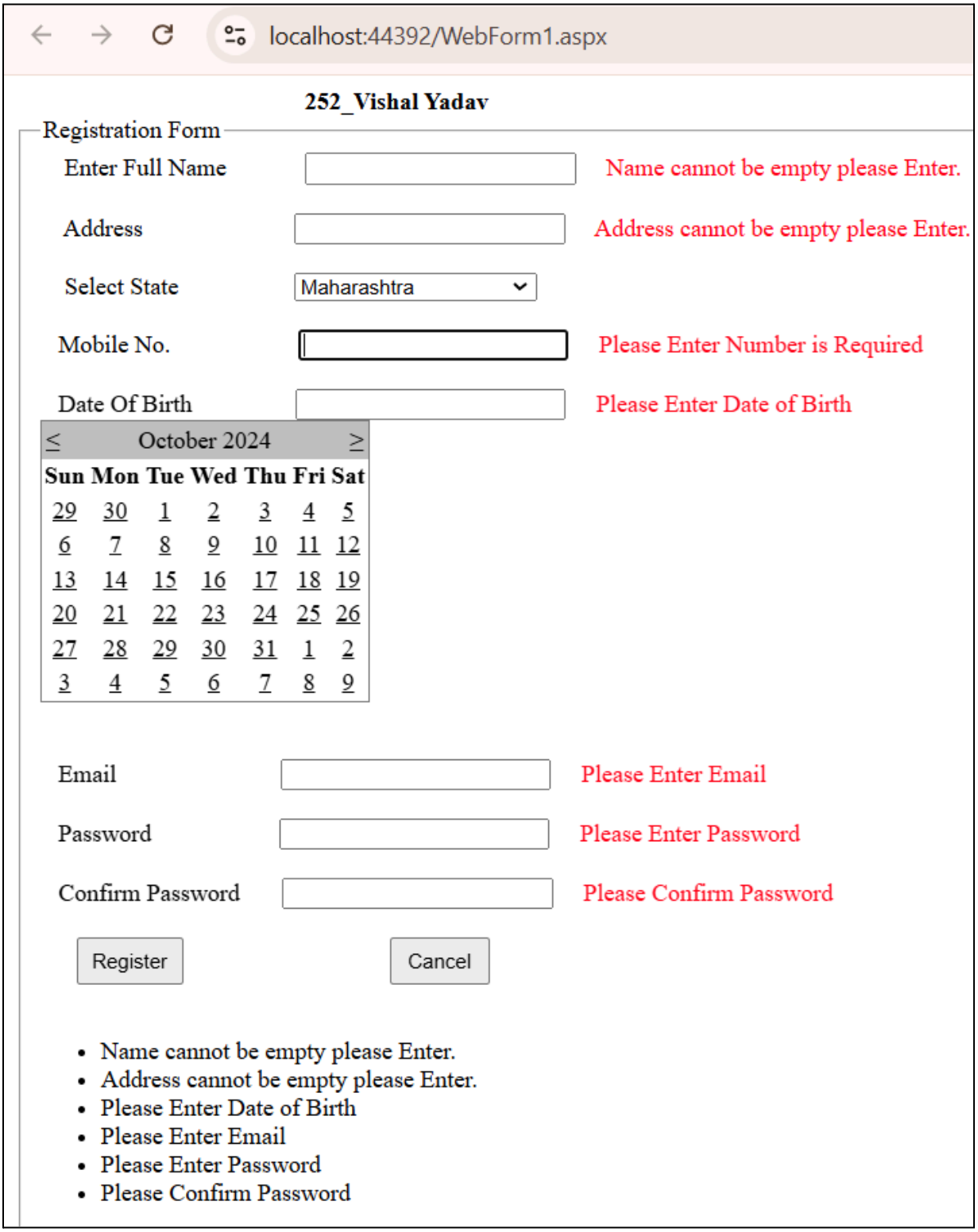
```
</body>
</html>
```

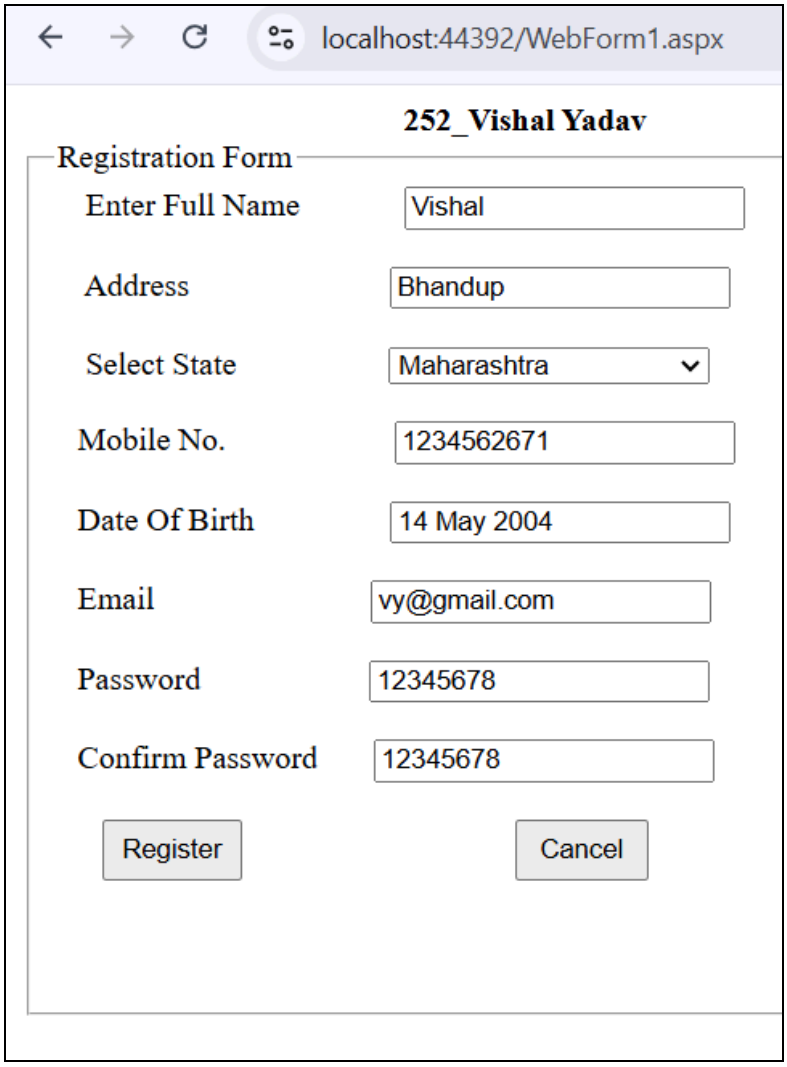
Webform3.aspx.cs

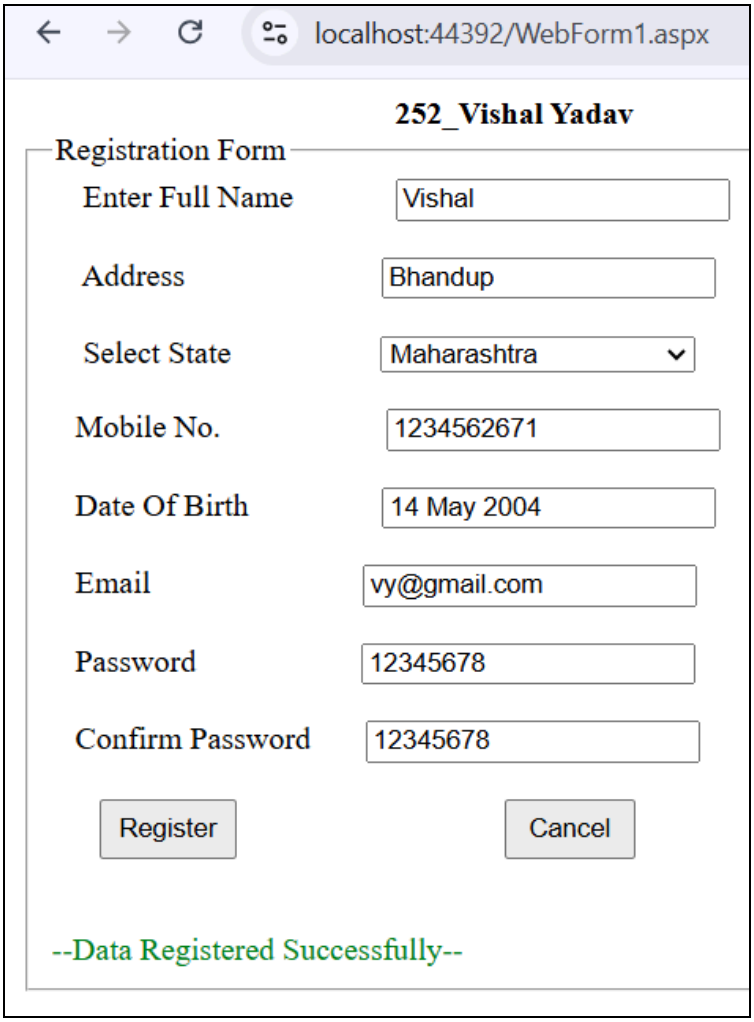
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace Vishal_P1
{
   public partial class WebForm3 : System.Web.UI.Page
   {
      protected void Page_Load(object sender, EventArgs e)
      {

      }

      protected void btn_submit_Click(object sender, EventArgs e)
      {
         if (RequiredFieldValidator1.IsValid == true &&
            RequiredFieldValidator2.IsValid == true &&
            RequiredFieldValidator3.IsValid == true &&
            RequiredFieldValidator4.IsValid == true &&
             RequiredFieldValidator5.IsValid == true &&
              RequiredFieldValidator6.IsValid == true &&
              RequiredFieldValidator7.IsValid == true &&

            RangeValidator1.IsValid == true &&
            CustomValidator1.IsValid == true &&
            RegularExpressionValidator1.IsValid == true &&
            CompareValidator1.IsValid == true)
         {
            Label.ForeColor = System.Drawing.Color.Green;
            Label.Text = "--Data Registered Successfully--";
         }
         else
         {
```

```
            Label.ForeColor = System.Drawing.Color.Red;
            Label.Text = "--Please Enter Valid Credential--";
        }
    }
    protected void CustomValidator1_ServerValidate(object source,
ServerValidateEventArgs args)

    {
        int len = args.Value.Length;
        if (len >= 8 && len <= 15)
        {
            args.IsValid = true;
        }

        else
        {
            args.IsValid = false;
        }
    }

    protected void btn_cancel_Click(object sender, EventArgs e)
    {
        txt_name.Text = "";

        txt_addr.Text = "";
        txt_contact.Text = "";
        txt_dob.Text = "";
        txt_email.Text = "";
        txt_pwd.Text = "";
        txt_pwd_confirm.Text = "";
    }

    protected void Calendar1_SelectionChanged(object sender, EventArgs e)
    {
        txt_dob.Text = Calendar1.SelectedDate.ToLongDateString();
        Calendar1.Visible = false;
    }
  }
}
```

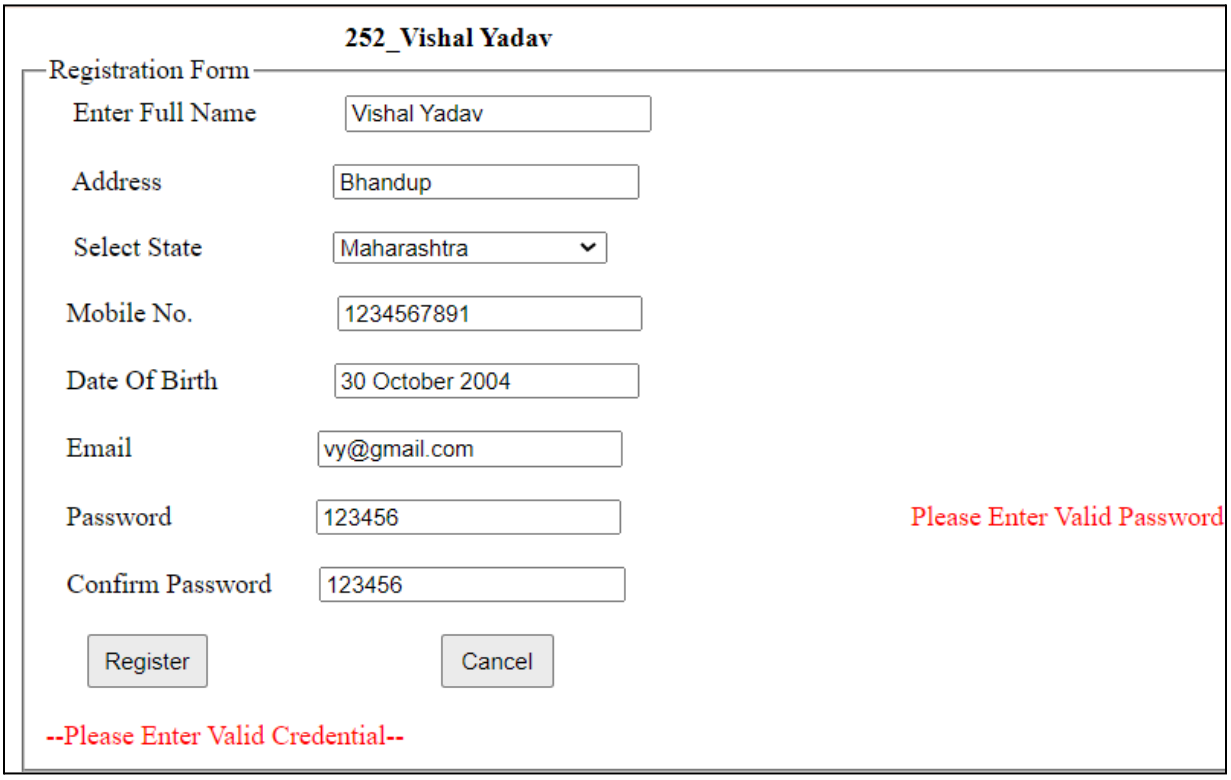
Output:-


localhost:44392/WebForm1.aspx
252_Vishal Yadav
Registration Form
Enter Full Name
Name cannot be empty please Enter.
Address
Address cannot be empty please Enter.
Select State
Maharashtra
Mobile No.
Please Enter Number is Required
Date Of Birth
Please Enter Date of Birth
October 2024
Sun Mon Tue Wed Thu Fri Sat
Email
Please Enter Email
Password
Please Enter Password
Confirm Password
Please Confirm Password
Register
Cancel
Name cannot be empty please Enter.
Address cannot be empty please Enter.
Please Enter Date of Birth
Please Enter Email
Please Enter Password
Please Confirm Password


localhost:44392/WebForm1.aspx
252_Vishal Yadav
Registration Form
Enter Full Name
Vishal
Address
Bhandup
Select State
Maharashtra
Mobile No.
1234562671
Date Of Birth
14 May 2004
Email
vy@gmail.com
Password
12345678
Confirm Password
12345678
Register
Cancel


localhost:44392/WebForm1.aspx
252_Vishal Yadav
Registration Form
Enter Full Name
Vishal
Address
Bhandup
Select State
Maharashtra
Mobile No.
1234562671
Date Of Birth
14 May 2004
Email
vy@gmail.com
Password
12345678
Confirm Password
12345678
Register
Cancel
--Data Registered Successfully--

Error Message



Conclusion:-
This example shows how to use ASP.NET's validation controls in a registration form to ensure proper input. Controls like RequiredFieldValidator, RangeValidator, RegularExpressionValidator, and CompareValidator verify that fields are correctly filled, such as ensuring a 10-digit contact number and matching passwords. Custom validation checks password length, while a ValidationSummary consolidates error messages. This approach enhances data accuracy and security, providing users with clear, real-time feedback.

10. Create a website using the master page concept.

Aim:-
To create a responsive and cohesive web application that utilizes a master page for consistent design and easy maintenance.

Theory:-
The master page concept allows developers to define a common layout (header, footer, and navigation) that can be reused across multiple web pages. This promotes a uniform user experience and simplifies updates—changes to the master page automatically propagate to all linked pages.

Objective:-
- To implement a master page for seamless navigation and layout.
- To enhance user experience by maintaining consistency across web pages.
- To reduce development time and effort through reusable components.

Code:-

Site2.master

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeBehind="Site2.master.cs" Inherits="Vishal_252.Site2" %>

<!DOCTYPE html>

<html>
<head runat="server">
  <title></title>
  <link rel="stylesheet" href="my.css" type="text/css" />
  <asp:ContentPlaceHolder ID="head" runat="server">
  </asp:ContentPlaceHolder>
</head>
<body>
  <header id="header">
<h1>252_Vishal Yadav</h1>
</header>
<nav id="nav">
```

```
  <ul>
    <li><a href="WebForm6.aspx">Home</a></li>
    <li><a href="WebForm7.aspx">About</a></li>
    <li><a href="WebForm8.aspx">Article</a></li>
    <li><a href="WebForm9.aspx">Contact</a></li>
  </ul>
</nav>
<aside id="side">
  <h1>news</h1>
  <a href="#"><p>creating html website</p></a>
  <a href="#"><p>learn css</p></a>
  <a href="#">learn c#</a>
</aside>


  <div id="con">
    <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">

    </asp:ContentPlaceHolder>
  </div>


<footer id="footer">
  copyright @252 Vishal Yadav
</footer>
</body>
</html>
</html>
```

WebForm6.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site2.Master"
AutoEventWireup="true" CodeBehind="WebForm6.aspx.cs"
Inherits="Vishal_252.WebForm6" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
runat="server">
  <h1>Home Page</h1>
</asp:Content>
```

WebForm7.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site2.Master"
AutoEventWireup="true" CodeBehind="WebForm7.aspx.cs"
Inherits="Vishal_252.WebForm7" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
runat="server">
  <h1>About Page</h1>
</asp:Content>
```

WebForm8.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site2.Master"
AutoEventWireup="true" CodeBehind="WebForm8.aspx.cs"
Inherits="Vishal_252.WebForm8" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
runat="server">
  <h1>Article Page</h1>
</asp:Content>
```

WebForm9.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site2.Master"
AutoEventWireup="true" CodeBehind="WebForm9.aspx.cs"
Inherits="Vishal_252.WebForm9" %>
<asp:Content ID="Content1" ContentPlaceHolderID="head"
runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder1"
runat="server">
  <h1>Contact</h1>
</asp:Content>
```

My.css-

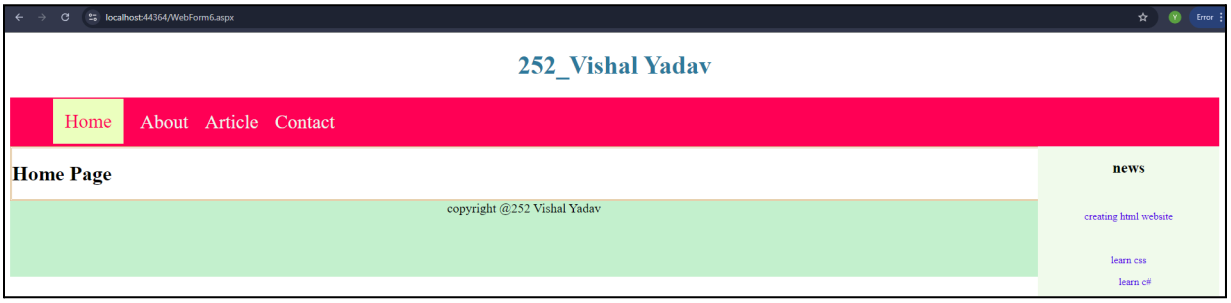
```
#header {
  color: #247BA0;
  text-align: center;
  font-size: 20px;
}
```

```
#nav {
   background-color: #FF1654;
   padding: 5px;
}

ul {
   list-style-type: none;
}

li a {
   color: #F1FAEE;
   font-size: 30px;
   column-width: 5%;
}

li {
   display: inline;
   padding-left: 2px;
   column-width: 20px;
}

a {
   text-decoration: none;
   margin-left: 20px
}

li a:hover {
   background-color: #F3FFBD;
   color: #FF1654;
   padding: 1%;
}

#side {
   text-align: center;
   float: right;
   width: 15%;
   padding-bottom: 79%;
   background-color: #F1FAEE;
}

#article {
   background-color: #EEF5DB;
```

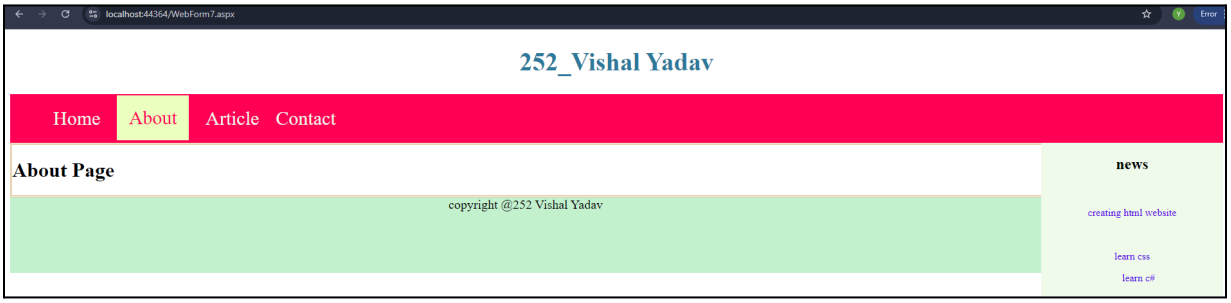
```
  padding: 10px;
  padding-bottom: 75%;
}

#footer {
  background-color: #C7EFCF;
  text-align: center;
  padding-bottom: 5%;
  font-size: 20px;
}

#con {
  border: double;
  border-color: burlywood;
}
```

Output:-

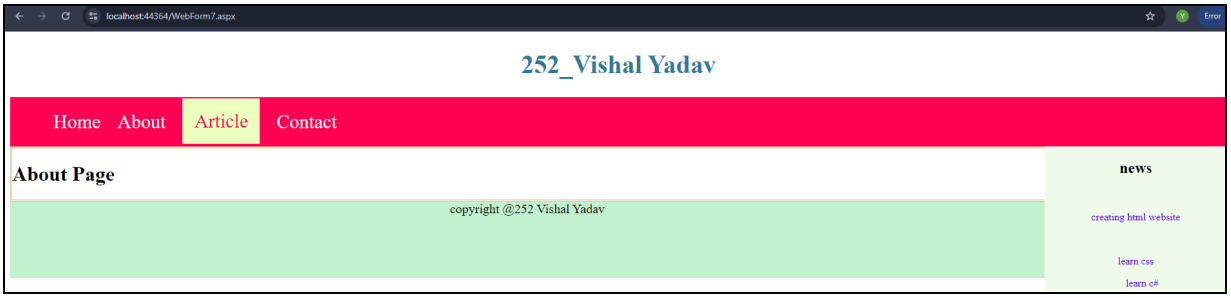
Home Page-



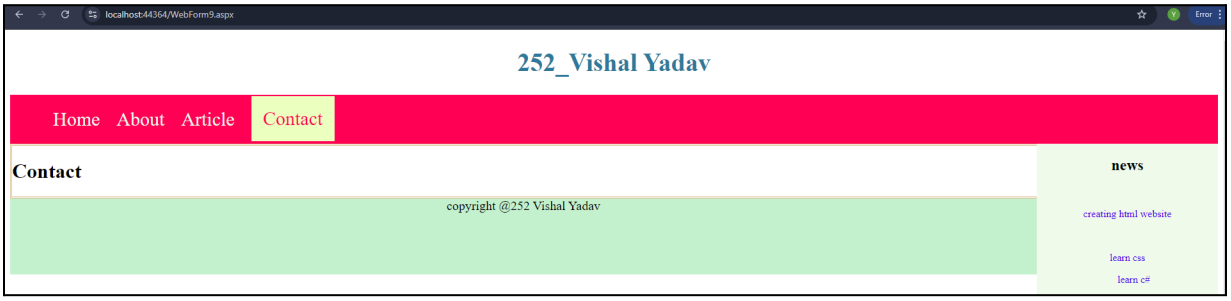
About Page-



Article Page-



Contact Page-



Conclusion:-

Utilizing the master page concept streamlines the web development process, ensures design consistency, and enhances maintainability. This approach not only improves the user experience but also allows developers to focus on content rather than repetitive design tasks.