**Practical Number 3: Design a Web Application using advanced controls.**

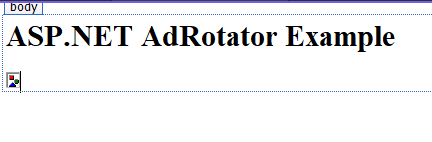
**1. AdRotator Control in ASP.NET**

The **AdRotator** control displays rotating advertisements based on an XML file.

**Steps to Run:**

1. Open Visual Studio and create an **ASP.NET Web Application**.
2. Add a new **Web Form** (AdRotator.aspx).
3. Create an XML file (Ads.xml) in the project.
4. Place some ad images in the Images folder.

**AdRotator.aspx**

****

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs" Inherits="Practical\_Number\_3.WebForm1" %>

<!DOCTYPE html>

<html>

<head>

<title>AdRotator Example</title>

</head>

<body>

<h2>ASP.NET AdRotator Example</h2>

<asp:AdRotator ID="AdRotator1" runat= "server" AdvertisementFile="~/Ads.xml" />

</body>

</html>

**Ads.xml (Advertisement File)**

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

<Advertisements>

<Ad>

<ImageUrl>Images/ad1.jpg</ImageUrl>

<NavigateUrl>https://www.example2.com</NavigateUrl>

<AlternateText>First Ad</AlternateText>

<Impressions>50</Impressions>

</Ad>

<Ad>

<ImageUrl>Images/ad2.jpg</ImageUrl>

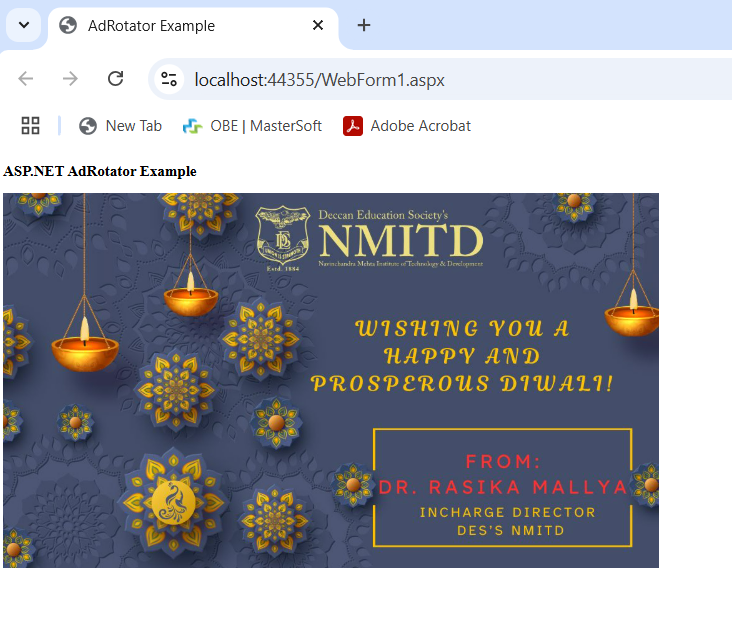
<NavigateUrl>https://www.example1.com</NavigateUrl>

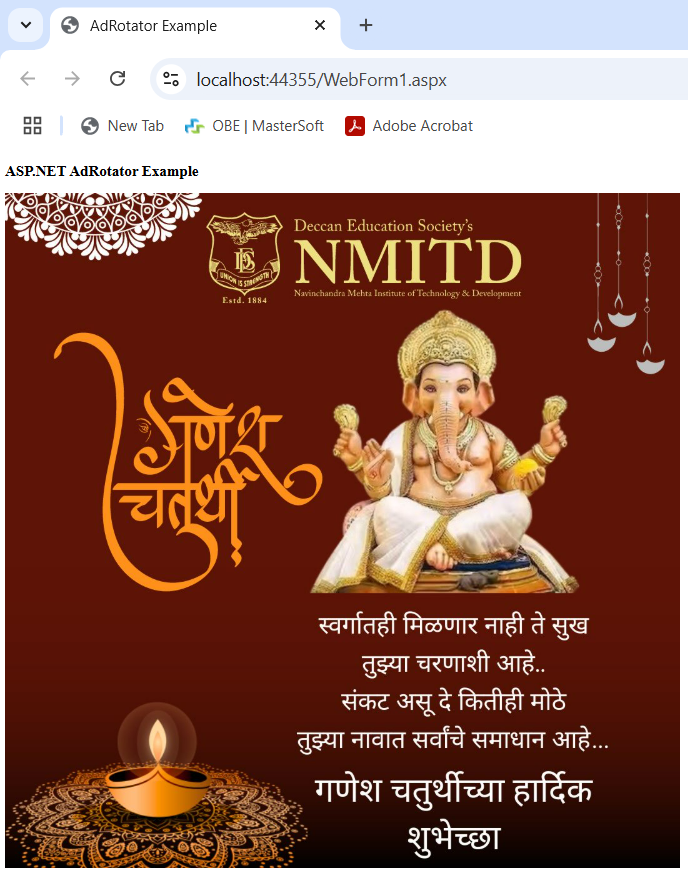
<AlternateText>Second Ad</AlternateText>

<Impressions>30</Impressions>

</Ad>

</Advertisements>





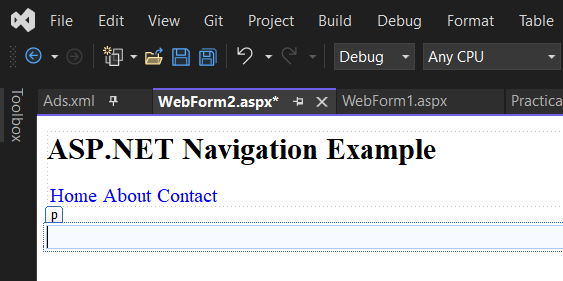
**2. Navigation Control (Menu Navigation)**

This example demonstrates how to use a **Menu control** for site navigation.

**Steps to Run:**

1. Create a new Web Form (Navigation.aspx).
2. Use the Menu control inside an asp:SiteMapDataSource.

**Navigation.aspx**

****

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs" Inherits="Practical\_Number\_3.WebForm2" %>

<!DOCTYPE html>

<html>

<head runat="server">

<title>Navigation Example</title>

</head>

<body>

<form id="form1" runat="server">

<h2>ASP.NET Navigation Example</h2>

<asp:Menu ID="Menu1" runat="server" Orientation="Horizontal">

<Items>

<asp:MenuItem Text="Home" NavigateUrl="Home.aspx"/>

<asp:MenuItem Text="About" NavigateUrl="About.aspx"/>

<asp:MenuItem Text="Contact" NavigateUrl="Contact.aspx"/>

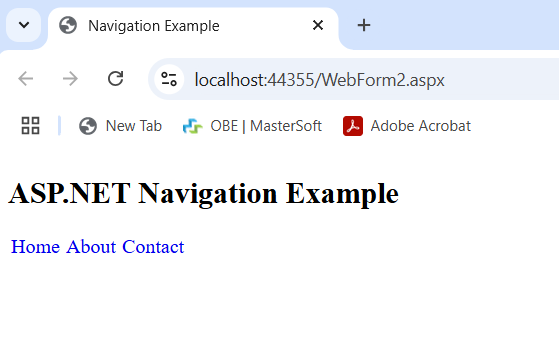
</Items>

</asp:Menu>

</form>

</body>

</html>



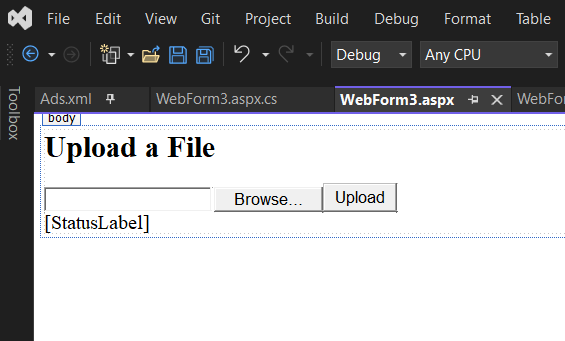
**3. File Upload in ASP.NET**

This example allows users to upload a file to the server.

**Steps to Run:**

1. Create a Web Form (FileUpload.aspx).
2. Implement the file upload functionality in C#.

**FileUpload.aspx**

****

<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm3.aspx.cs" Inherits="Practical\_Number\_3.WebForm3" %>

<!DOCTYPE html>

<html>

<head runat="server">

<title>File Upload Example</title>

</head>

<body>

<form id="form1" runat="server">

<h2>Upload a File</h2>

<asp:FileUpload ID="FileUploadControl" runat="server" />

<asp:Button ID="UploadButton" runat="server" Text="Upload" OnClick="UploadButton\_Click" />

<br />

<asp:Label ID="StatusLabel" runat="server" Text=""></asp:Label>

</form>

</body>

</html>

**FileUpload.aspx.cs**

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Web;

using System.Web.UI;

using System.Web.UI.WebControls;

namespace Practical\_Number\_3

{

public partial class WebForm3 : System.Web.UI.Page

{

protected void Page\_Load(object sender, EventArgs e)

{

}

protected void UploadButton\_Click(object sender, EventArgs e)

{

if (FileUploadControl.HasFile)

{

try

{

string filename = Path.GetFileName(FileUploadControl.FileName);

FileUploadControl.SaveAs(Server.MapPath("~/Uploads/") + filename);

StatusLabel.Text = "Upload status: File uploaded successfully!";

}

catch (Exception ex)

{

StatusLabel.Text = "Upload status: Error - " + ex.Message;

}

}

else

{

StatusLabel.Text = "Upload status: No file selected.";

}

}

}

}

