PIEAS

Pakistan Institute of Engineering and Applied Sciences

Internet Application Development

Report: Lab No 2

Demonstration of ASP.NET Page Processing

Submitted To

Dr. Irfan Hameed

Submitted By

Insha Afzal

DEPARTMENT: CIS(22-26)

Develop a page to demonstrate various events which are raised and handled during ASP.NET Page processing once the page object has been created? Develop a story about it?

Coding Section:

Web.config

```xml
<?xml version="1.0"?>
<configuration>
        <!--
   For a description of web.config changes see
http://go.microsoft.com/fwlink/?LinkId=235367.

   The following attributes can be set on the <httpRuntime> tag.
    <system.Web>
     <httpRuntime targetFramework="4.8.1" />
    </system.Web>
  -->
        <system.web>
                <compilation debug="true" targetFramework="4.8.1"/>
                <pages controlRenderingCompatibilityVersion="4.0"/>
        </system.web>
</configuration>
```

Default.aspx

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Default.aspx.vb"
Inherits="_Default" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>ASP.NET Page Processing Events </title>
</head>
<body>
   <form id="form1" runat="server">
     <div>
        <asp:Button ID="btnClickMe" runat="server" Text="Click Me!"
OnClick="btnClickMe_Click" />
     </div>
   </form>
```

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

Default.aspx.vb

```vb
Partial Class _Default
    Inherits System.Web.UI.Page

    Protected Overrides Sub OnInit(ByVal e As EventArgs)
        MyBase.OnInit(e)
        Response.Write("<h2>ASP.NET Page Processing Events Demo</h2>")
        Response.Write("<p>This page demonstrates the various events raised and handled
during the ASP.NET Page Processing lifecycle.</p><br/>")

        Response.Write("Controls are initialized.<br/>")
    End Sub

    Protected Overrides Sub LoadViewState(ByVal savedState As Object)
        MyBase.LoadViewState(savedState)
        Response.Write("View state loaded.<br/>")
    End Sub

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
        Response.Write("Page is loaded.<br/>")
    End Sub

    Protected Sub btnClickMe_Click(ByVal sender As Object, ByVal e As EventArgs)
        Response.Write("Button was clicked.<br/>")
    End Sub

    Protected Overrides Function SaveViewState() As Object
        Response.Write("Saving view state.<br/>")
        Return MyBase.SaveViewState()
    End Function

    Protected Overrides Sub Render(ByVal writer As HtmlTextWriter)
        Response.Write("Page is being rendered.<br/>")
        MyBase.Render(writer)
    End Sub

    Protected Sub Page_Unload(ByVal sender As Object, ByVal e As EventArgs) Handles
```
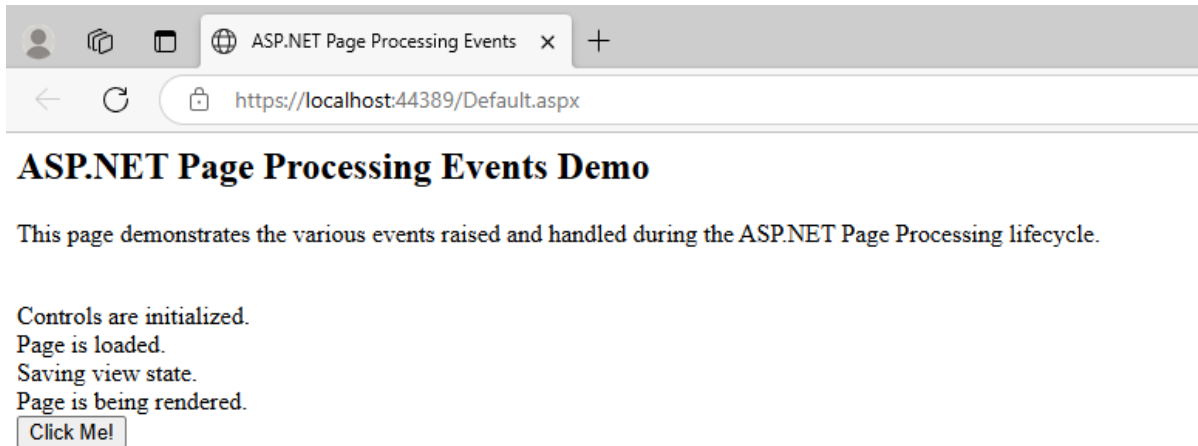
```
    Me.Unload
        ' Cannot use Response.Write here
        System.Diagnostics.Debug.WriteLine("Page is being unloaded.<br/>")
    End Sub
End Class
```

Output:

"The Journey of an ASP.NET Page"

1. **Backstage Preparation: OnInit**
   When a page request arrives, the backstage crew (ASP.NET) gets ready by initializing the page controls. During this phase, the **OnInit** event fires, preparing all elements for display. The message "Controls are initialized" lets us know that everything is set for the show!

2. **Preparing the Performance: LoadViewState**
   Just before the performance begins, ASP.NET ensures the page appears as it did during previous visits by loading the **ViewState**. The message "View state loaded" confirms that all previous settings, like user inputs or selections, are preserved.

3. **The Performance Begins: Page_Load**
   The curtain rises, and the page starts loading. The **Page_Load** event is triggered, and we see the message "Page is loaded." This is when the page's dynamic content is generated and displayed.

4. **Audience Interaction: Button Click**
   Now, imagine the audience interacting with the performance. When the user clicks the "Click Me!" button, the **btnClickMe_Click** event is triggered, displaying the message "Button was clicked." ASP.NET reacts to this interaction, similar to how a performer might respond to audience engagement.

5. **Saving the Show: SaveViewState**
   Before the performance ends, ASP.NET saves any changes or interactions in the

ViewState. The message "Saving view state" reminds us that everything is recorded so the next time the page is visited, the experience remains consistent.
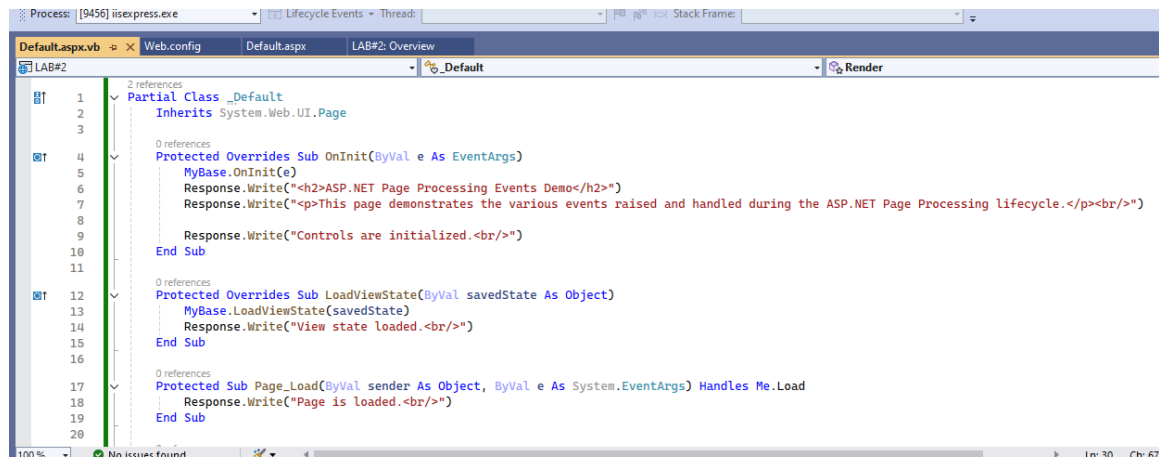
6. **The Final Act: Render**
As the page is ready to be delivered to the browser, the **Render** event is triggered, finalizing the show. The page is then rendered to the screen, and the message "Page is being rendered" lets us know that it's all set for the user to see.
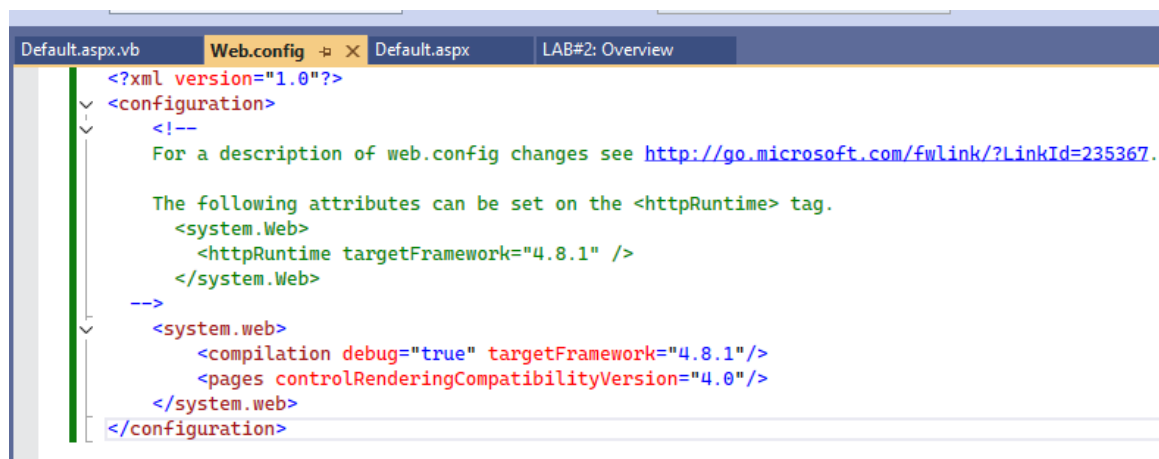
7. **The Curtain Falls: Page_Unload**
After the performance, ASP.NET cleans up resources. The **Page_Unload** event is called when the page is unloaded, ensuring that any necessary cleanup happens in the background (though this doesn't display on the page).

Code snippets:


```vb
Partial Class _Default
    Inherits System.Web.UI.Page

    Protected Overrides Sub OnInit(ByVal e As EventArgs)
        MyBase.OnInit(e)
        Response.Write("<h2>ASP.NET Page Processing Events Demo</h2>")
        Response.Write("<p>This page demonstrates the various events raised and handled during the ASP.NET Page Processing lifecycle.</p><br/>")

        Response.Write("Controls are initialized.<br/>")
    End Sub

    Protected Overrides Sub LoadViewState(ByVal savedState As Object)
        MyBase.LoadViewState(savedState)
        Response.Write("View state loaded.<br/>")
    End Sub

    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        Response.Write("Page is loaded.<br/>")
    End Sub
```



```xml
<?xml version="1.0"?>
<configuration>
    <!--
    For a description of web.config changes see http://go.microsoft.com/fwlink/?LinkId=235367.

    The following attributes can be set on the <httpRuntime> tag.
      <system.Web>
        <httpRuntime targetFramework="4.8.1" />
      </system.Web>
    -->
    <system.web>
        <compilation debug="true" targetFramework="4.8.1"/>
        <pages controlRenderingCompatibilityVersion="4.0"/>
    </system.web>
</configuration>
```

```aspx
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="Default.aspx.vb" Inherits="_Default" %>

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>ASP.NET Page Processing Events </title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <asp:Button ID="btnClickMe" runat="server" Text="Click Me!" OnClick="btnClickMe_Click" />
        </div>
    </form>
</body>
</html>
```