Sample Project using Code-Behind

Here is an example of the same guestbook application being coded using code-behind classes:

```
using System;
using System.Data;
using System.Data.OleDb;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Web.UI.HtmlControls;
namespace Chapter2.CodeBehindStyle
{
    public partial class Default : System.Web.UI.Page
    {
        protected void btnComments_Click(object sender, EventArgs e)
        {
            LoadComments();
        }
        /// <summary>
        /// Load all comments from the Access DB
        /// </summary>
        private void LoadComments()
        {
            string AppPath = System.AppDomain.CurrentDomain.
                             BaseDirectory.ToString();
            string sCon = @"Provider=Microsoft.JET.OLEDB.4.0; Data
                     Source=" + AppPath + "/App_Data/Guestbook.mdb";
            using (OleDbConnection cn = new OleDbConnection(sCon))
            {
                string sQuery = @"SELECT * FROM Guestbook order by
                                  EntryDate desc";
                OleDbCommand cmd = new OleDbCommand(sQuery, cn);
                OleDbDataAdapter da = new OleDbDataAdapter(cmd);
                DataSet ds = new DataSet();
                cn.Open();
                da.Fill(ds);
                rptComments.DataSource = ds;
                rptComments.DataBind();
            }
        }
    }//end class
}//end namespace
```

In the code above we are simply loading all guestbook entries in the LoadComments() method and binding it to a repeater (rptComments) in the code-behind partial class file.

Here is the ASPX form:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Default.
aspx.cs" Inherits="Chapter2.CodeBehindStyle.Default" %>
<html>
<head>
    <title>Chapter 2: Code-behind sample in ASP.NET</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:Button ID="btnComments" runat="server" Text="View All
                          Comments" OnClick="btnComments_Click" />

         <h1>Guestbook Entries</h1>

       <asp:Repeater id="rptComments" runat="server">
            <ItemTemplate>
               Name:    <%# Eval("FullName")%>
                                       <br>
               Email:  <%# Eval("EmailID")%>
                                       <br>
               Website: <%# Eval("Website")%>
                                       <br>
               Dated:   <%# Eval("EntryDate")%>
                                       <br>
               Comments: <%# Eval("Comments")%>
            </ItemTemplate>
       </asp:Repeater>
    </div>
    </form>
</body>
</html>
```

As you can see, we don't have any C# or VB.NET coding in the ASPX pages; all of the managed code is in the code-behind class. Also note that the declaration of any server side control is put in a separate designer.cs file, which is auto-generated after parsing the ASPX markup file. So we have a clean logical separation of the declarative ASPX controls placement (HTML part) and the actual managed code in the code-behind partial classes. Here is a diagrammatic representation of this 1-tier 1-layer style having 2 sub-layers in the main UI layer: