State Management

Introduction

- The HTTP protocol is stateless. We need to use complementary state management techniques.

The ASP.NET View State

- The ASP.NET runtime environment automatically embeds a hidden form field (named `__VIEWSTATE`) that flows between the web browser and the server.
- The `System.Web.UI.Page` Init event handler is responsible for reading the incoming values in the `__VIEWSTATE` field and populating the appropriate member variables on the server side. In addition, just before the reply is sent back to the client the `__VIEWSTATE` data is used to repopulate the form's widgets.

The ASP.NET View State

- The page directive includes the `EnableViewState` attribute. Its default value is `true`. If we assign it with `false` then the view state mechanism won't work.

```
<%@ Page Title="" EnableViewState="false" Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="WebApplication20.WebForm1" %>
```

The ASP.NET View State

- The `System.Web.UI.Control` base class includes the definition of `ViewState`, a protected property that enables us to access a `System.Web.UI.StateBag` object that represents the data contained within the `__VIEWSTATE`.

The ASP.NET View State

- The `System.Web.UI.Control` base class defines the `ViewState` property.
- When referring this property we shall get access to a `System.Web.UI.StateBag` object that represents the data contained within the `__VIEWSTATE` field.

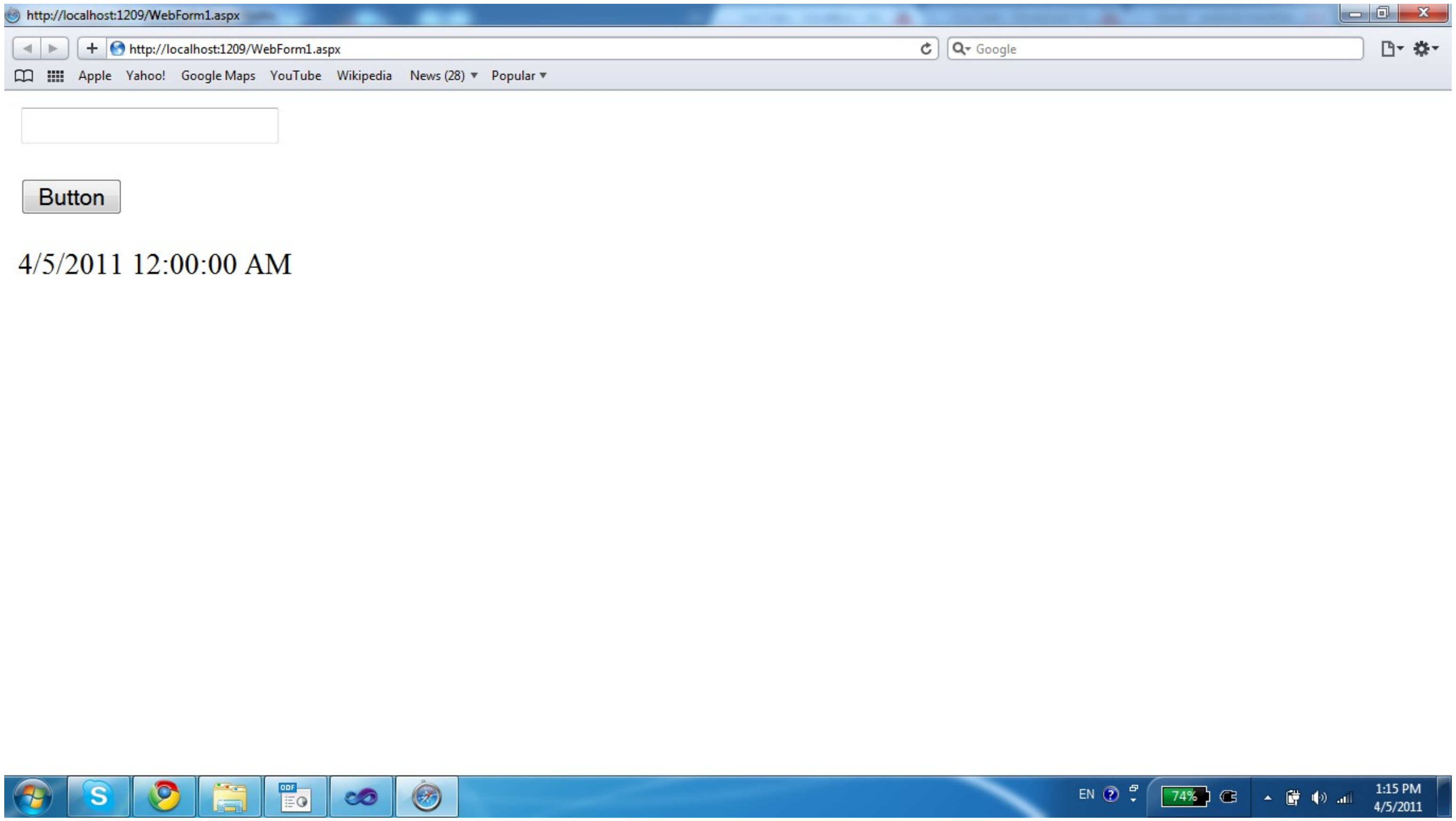
The ASP.NET View State

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication24.WebForm1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
    <body>
    <form id="form1" runat="server">
        <div>
        <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
        <p />
        <asp:Button ID="Button1" runat="server" Text="Button" />
        <p />
        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
        </div>
    </form>
</body>
</html>
```

The ASP.NET View State

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

namespace WebApplication24
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (ViewState["CreationTimeStamp"] == null)
            {
                ViewState["CreationTimeStamp"] = DateTime.Today.ToString();
            }
            Label1.Text = (string)ViewState["CreationTimeStamp"];
        }
    }
}
```

The ASP.NET View State



4/5/2011 12:00:00 AM

The `Global.asax` File

- We can easily add this file into our ASP.NET web application. The purpose of this file is to specify run-time behavior for our web application. Implementing our code within this file we will get event handlers that allow you to interact with the application and the session levels.
- The auto generated code includes the definition of the `Global` class, that extends `System.Web.HttpApplication`.

The `Global.asax` File

```
namespace WebApplication20
{
    public class Global : System.Web.HttpApplication
    {

        protected void Application_Start(object sender, EventArgs e)
        {

        }

        protected void Session_Start(object sender, EventArgs e)
        {

        }

        protected void Application_BeginRequest(object sender, EventArgs e)
        {

        }

        protected void Application_AuthenticateRequest(object sender, EventArgs e)
        {

        }
```

The `Global.asax` File

```
        protected void Application_Error(object sender, EventArgs e)
        {

        }

        protected void Session_End(object sender, EventArgs e)
        {

        }

        protected void Application_End(object sender, EventArgs e)
        {

        }
    }
}
```

Last Chance Exception Handler

- We can implement the `Application_Error()` function in order to set a last chance exception handler.
- This function will handle errors that weren't handled elsewhere. This function is the last place to handle an exception.
- We can get information about the exception that took place by accessing the `System.Exception` property.

Last Chance Exception Handler

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication25.WebForm1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
        <asp:Label ID="Label1" runat="server" Text="/"></asp:Label>
        <asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>
        <asp:Label ID="Label2" runat="server" Text="="></asp:Label>
        <asp:Label ID="Label3" runat="server" Text="..."></asp:Label>
        <p />
        <asp:Button ID="Calculate" runat="server" Text="Calculate"
             onclick="Calculate_Click" />
    </div>
    </form>
</body>
</html>
```

Last Chance Exception Handler

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

namespace WebApplication25
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            throw (new ArgumentNullException());
        }

        protected void Calculate_Click(object sender, EventArgs e)
        {
            double result = double.Parse(TextBox1.Text) /
                double.Parse(TextBox2.Text);
            Label3.Text = result.ToString();
        }
    }
}
```

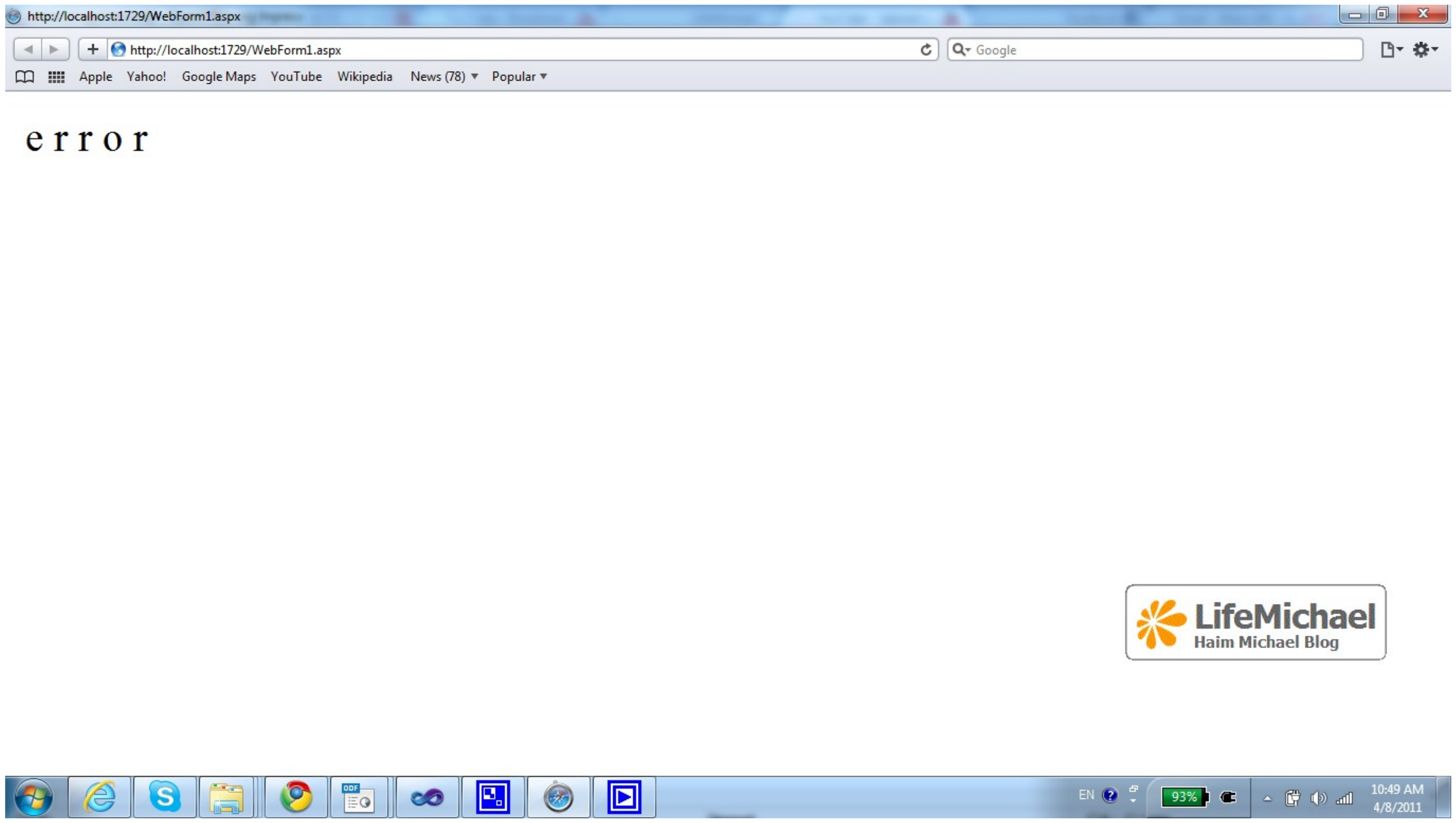
Last Chance Exception Handler

```
namespace WebApplication25
{
    public class Global : System.Web.HttpApplication
    {
        protected void Application_Start(object sender, EventArgs e)
        {
        }

        protected void Session_Start(object sender, EventArgs e)
        {
        }

        protected void Application_BeginRequest(object sender, EventArgs e)
        {
        }

        protected void Application_AuthenticateRequest(object sender, EventArgs e)
        {
        }
```

Last Chance Exception Handler

```
        protected void Application_Error(object sender, EventArgs e)
        {
            System.Web.UI.Page page = System.Web.HttpContext.Current.Handler
                as System.Web.UI.Page;
            ((Label)page.FindControl("Label3")).Text =
                Server.GetLastError().ToString();
            Server.ClearError();
        }

        protected void Session_End(object sender, EventArgs e)
        {
        }

        protected void Application_End(object sender, EventArgs e)
        {
        }
    }
}
```

Last Chance Exception Handler



```
error
```

The `HttpApplication` Class

- This class includes the definition for important members worth knowing.
- The `Application` property returns a reference for an `HttpApplicationState` object, we can use for maintaining application level data.
- The `Request` property returns a reference for an `HttpRequest` object that represents the HTTP request.
- The `Response` property returns a reference for an `HttpResponse` object that represents the HTTP response.

The `HttpApplication` Class

- The `Server` property returns a reference for an `HttpServerUtility` object, we can use for maintaining server level data.
- The `Session` property returns a reference for an `HttpSessionState` object that represents the current session. We can use this object for maintaining session level data.

Maintaining Session Data

- Each one of the users has his own session data. Each user's session data represents the ongoing interaction with a web application.
- The `HttpSessionState` type allows us to maintain stateful data for a specific user. Each `HttpSessionState` object holds the session data of a specific user. The `HttpSessionState` includes the definition for a simple indexer we can use for setting and getting session data.

Maintaining Session Data

- The code we place within the `Global.asax` file allows us to intercept the beginning and the ending of every session by adding our code into the `Session_Start()` and the `Session_End()` events methods.
- We can modify the `Session_Start()` method in order to create per user data items.
- We can modify the `Session_End()` method in order to perform any work that needs to be done (e.g. freeing none managed resources) when the session ends.

Maintaining Session Data

- Similarly to application level data, when managing session level data we can hold any `System.Object` derived type.

```
public void Session_Start(Object sender, EventsArgs e)
{
   Session[“cart”] = new ShoppingCart();
}

public void Session_End(Object sender, EventsArgs e)
{
   //saving shopping cart data to database
   Session.RemoveAll();
}
```

Maintaining Session Data

- The `Page.Session` property allows us to maintain session data in between each request the user send to the server.

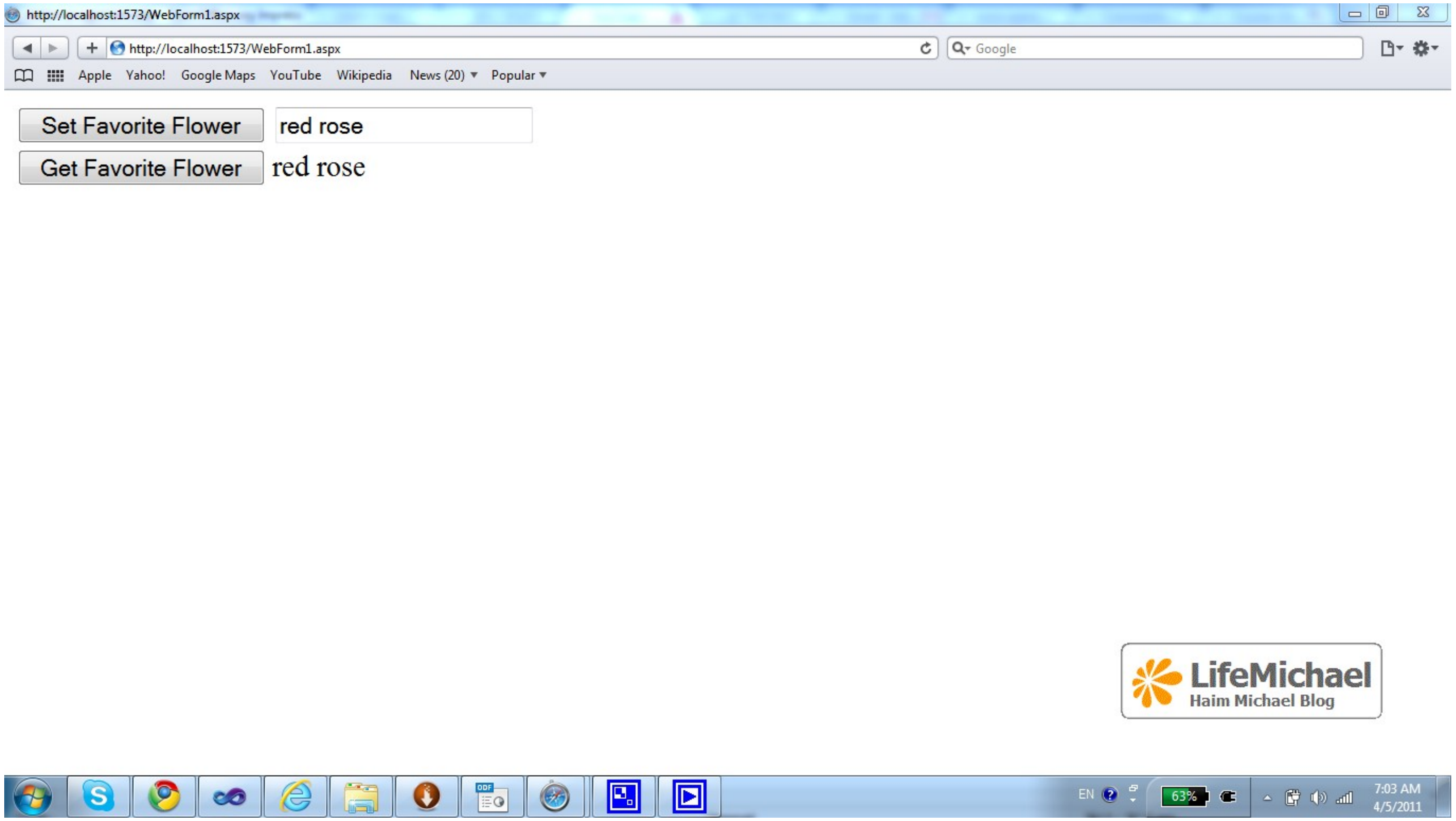
Maintaining Session Data

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication23.WebForm1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:Button ID="Button1" runat="server" Text="Set Favorite Flower"
            onclick="Button1_Click" Width="135px" />
        <asp:TextBox ID="TextBox1" runat="server" />
    </div>
    <div>
        <asp:Button ID="Button2" runat="server" Text="Get Favorite Flower"
            onclick="Button2_Click" Width="135px" />
        <asp:Label ID="Label1" runat="server" Text="Label"></asp:Label>
    </div>
    </form>
</body>
</html>
```

Maintaining Session Data

```
namespace WebApplication23
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        //private string favorite = "rose";
        protected void Page_Load(object sender, EventArgs e)
        {
            //this.Label1.Text = favorite;
            this.Label1.Text = (String)Session["flower"];
        }
        protected void Button1_Click(object sender, EventArgs e)
        {
            //favorite = this.TextBox1.Text;
            Session["flower"] = this.TextBox1.Text;
        }
        protected void Button2_Click(object sender, EventArgs e)
        {
            //Label1.Text = favorite;
            Label1.Text = (String)Session["flower"];
        }
    }
}
```

Maintaining Session Data



Set Favorite Flower red rose

Get Favorite Flower red rose

Maintaining Application Data

- We can access the `HttpApplicationState` object from within the scope of our page code behind. Referring the `Application` property returns a reference for the `HttpApplicationState` object that represents our web application.

YouTube

Maintaining Application Data

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.SessionState;

namespace WebApplication26
{
    public class CurrenciesData
    {
        public double Usd = 3.5;
        public double Euro = 4.9;
        public double Gbp = 5.9;
    }
    public class Global : System.Web.HttpApplication
    {

        protected void Application_Start(object sender, EventArgs e)
        {
            Application["currencies"] = new CurrenciesData();
        }
```

Maintaining Application Data

```
        protected void Session_Start(object sender, EventArgs e)
        {
        }

        protected void Application_BeginRequest(object sender, EventArgs e)
        {
        }

        protected void Application_AuthenticateRequest(object sender, EventArgs e)
        {
        }

        protected void Application_Error(object sender, EventArgs e)
        {
        }

        protected void Session_End(object sender, EventArgs e)
        {
        }

        protected void Application_End(object sender, EventArgs e)
        {
        }
    }
}
```

Maintaining Application Data

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication26.WebForm1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        USD exchange rate is <asp:Label ID="Label1"
            runat="server" Text="Label"></asp:Label>
    </div>
    </form>
</body>
</html>
```

Maintaining Application Data

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

namespace WebApplication26
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            this.Label1.Text =
                ((CurrenciesData)Application["currencies"]).Usd.ToString();
        }
    }
}
```

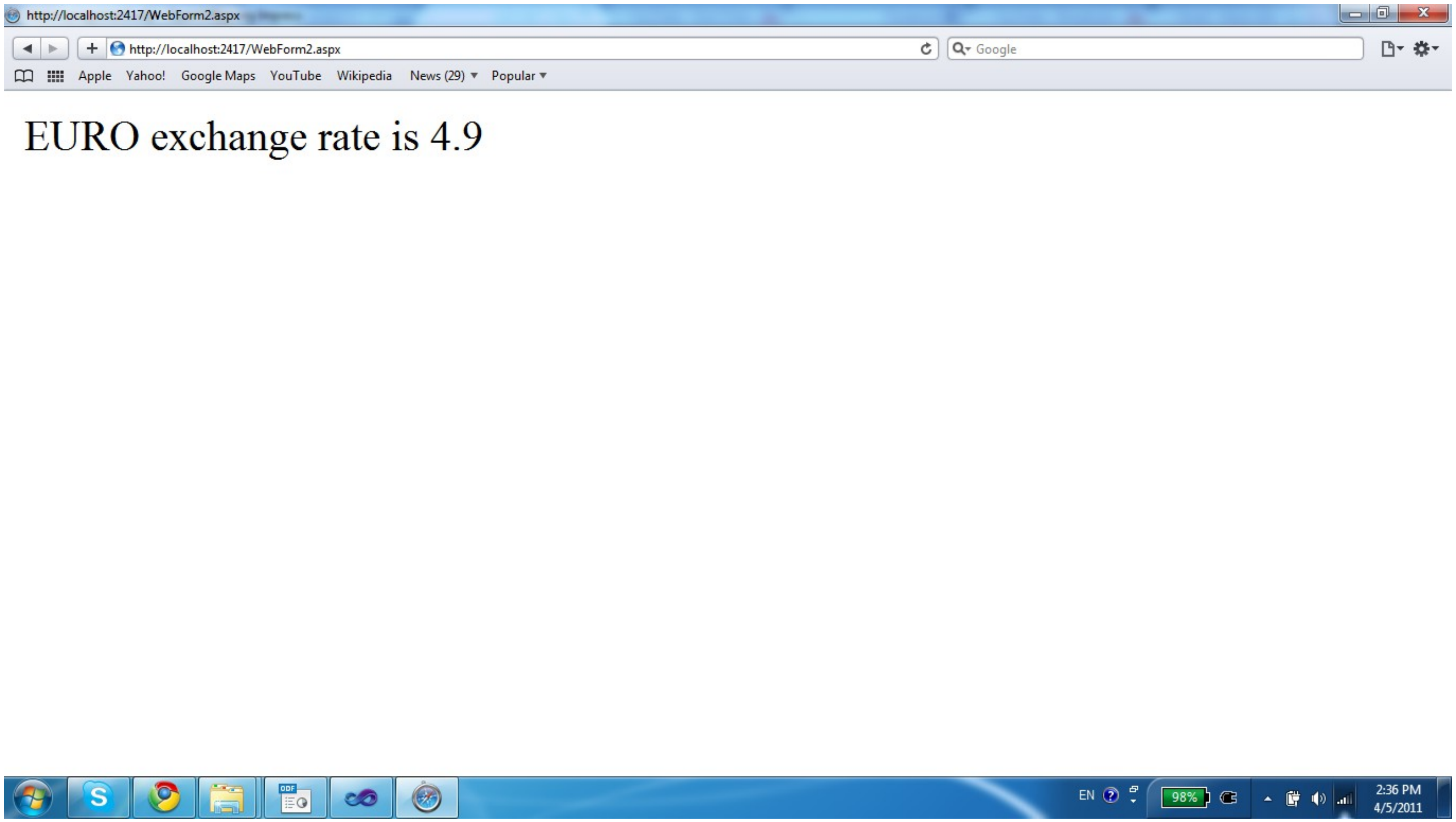
Maintaining Application Data

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm2.aspx.cs"
Inherits="WebApplication26.WebForm2" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        EURO exchange rate is <asp:Label ID="Label1" runat="server"
            Text="Label"></asp:Label>
    </div>
    </form>
</body>
</html>
```

Maintaining Application Data

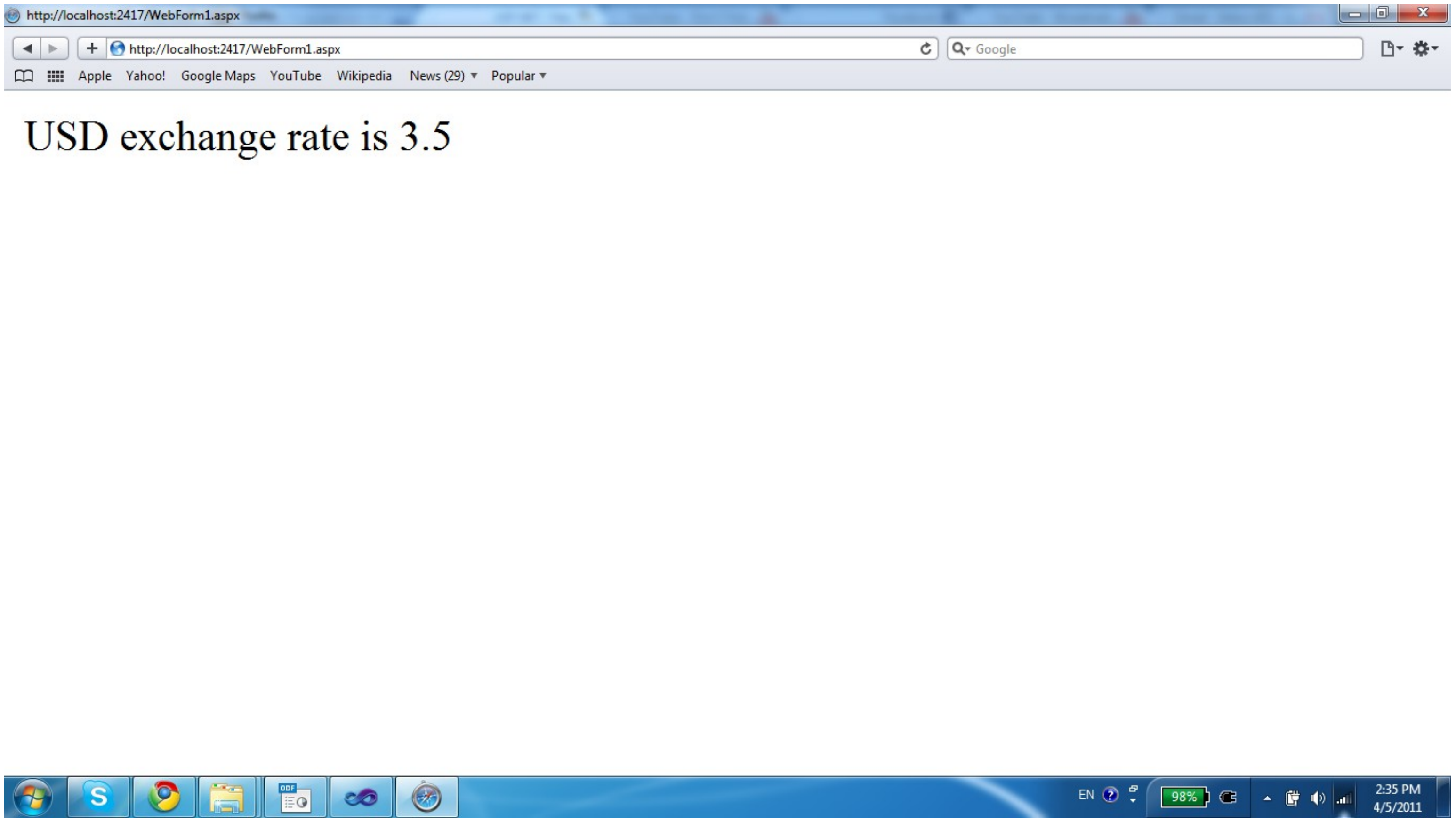
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace WebApplication26
{
    public partial class WebForm2 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            Label1.Text =
                ((CurrenciesData)Application["currencies"]).Euro.ToString();
        }
    }
}
```

Maintaining Application Data



```
EURO exchange rate is 4.9
```

Maintaining Application Data



```
USD exchange rate is 3.5
```

Maintaining Application Data

- When setting more than one application variable at the same time we should consider to lock on the `Application` object in order to prevent data corruption.

```
//safely accessing application data
Application.Lock();
Application["inventory"] = new Inventory();
Applicaiton["tariff"] = new Tariff();
…
Application.UnLock();
```

Application Cache

- When we need to keep application data for no longer than a limited period of time we can avoid the usage of the `HttpApplicationState` object. We can use the `System.Web.Caching.Cache` object instead.
- Interacting with the `Cache` object is similar to the way we interact with the `HttpApplicationState` object.
- We can access the `Cache` object using the `HttpContext.Cache` property.

Application Cache

- The auto generated `Global` class extends the `System.Web.HttpApplication` class. The later extends `HttpContext`. Therefore, within the scope of `Global.asax` we can access the cache using the `Context` property.
- Within the scope of a page we can access the cache using the `Cache` property.

```
...
System.Web.Caching.Cache cache = Context.Cache;
cache["backgroundmusic"] = "airsupply";
...
```

Application Cache

- When calling the Add function on our cach object we can easily specify its expiration.

```
public Object Add(
    string key,
    Object value,
    CacheDependency dependencies,
    DateTime absoluteExpiration,
    TimeSpan slidingExpiration,
    CacheItemPriority priority,
    CacheItemRemovedCallback onRemoveCallback
)
```

Cookies

- The `System.Web.HttpCookie` class on the server side represents the cookie that is eventually created in the client web browser.
- In order to create a new cookie on the client end we should instantiate `System.Web.HttpCookie` and add the new created object to the collection accessed through the `Response.Cookies` property.

Cookies

- In order to persist the cookie data back to the client hard drive make sure to explicitly set an expiration date using the `HttpCookies.Expires` property. Unless we explicitly set an expiration date the cookie will be deleted when the user shut down the browser.
- We can access all cookies that came along together with the request by referring the `Request.Cookies` collection.

The Session State Server

- We can instruct the ASP.NET runtime environment to host the session state *.dll in a separated process known as the ASP.NET session state server (aspnet_state.exe).
- This separated process can be executed on a separated server. When executing it on a separated server we should update the Web.Config configuration file.
- We can configure this server to keep all session data saved in a database.

Sample

- The following sample presents a simple possible code you can use for getting some inspiration when developing a web store. It is a proof of concept sample only. There are many parts we can improve. For the sake of simplicity and for the sake of keeping this sample as short as possible I chose to code the simplest sample I had in mind.

Sample

The Global Class

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Security;
using System.Web.SessionState;

namespace SimpleWebStore
{
    public class Global : System.Web.HttpApplication
    {

        protected void Application_Start(object sender, EventArgs e)
        {
            Application["tariff"] = new Tariff();
        }

        protected void Session_Start(object sender, EventArgs e)
        {
            Session["cart"] = new ShoppingCart();
        }
```

Sample

```
        protected void Application_BeginRequest(object sender, EventArgs e)
        {
        }

        protected void Application_AuthenticateRequest(object sender, EventArgs e)
        {
        }

        protected void Application_Error(object sender, EventArgs e)
        {
        }

        protected void Session_End(object sender, EventArgs e)
        {
            //saving shopping cart data to the database
        }

        protected void Application_End(object sender, EventArgs e)
        {
        }
    }
}
```

Sample

The Tariff,ShoppingCart, ShoppingCartLine and Product Classes

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

namespace SimpleWebStore
{
    public class Tariff
    {
        private IDictionary<int,double> prices =
            new Dictionary<int,double>();
        public double this[int id]
        {
            get
            {
               return prices[id];
            }
            set
            {
                prices[id] = value;
            }
        }
    }
```

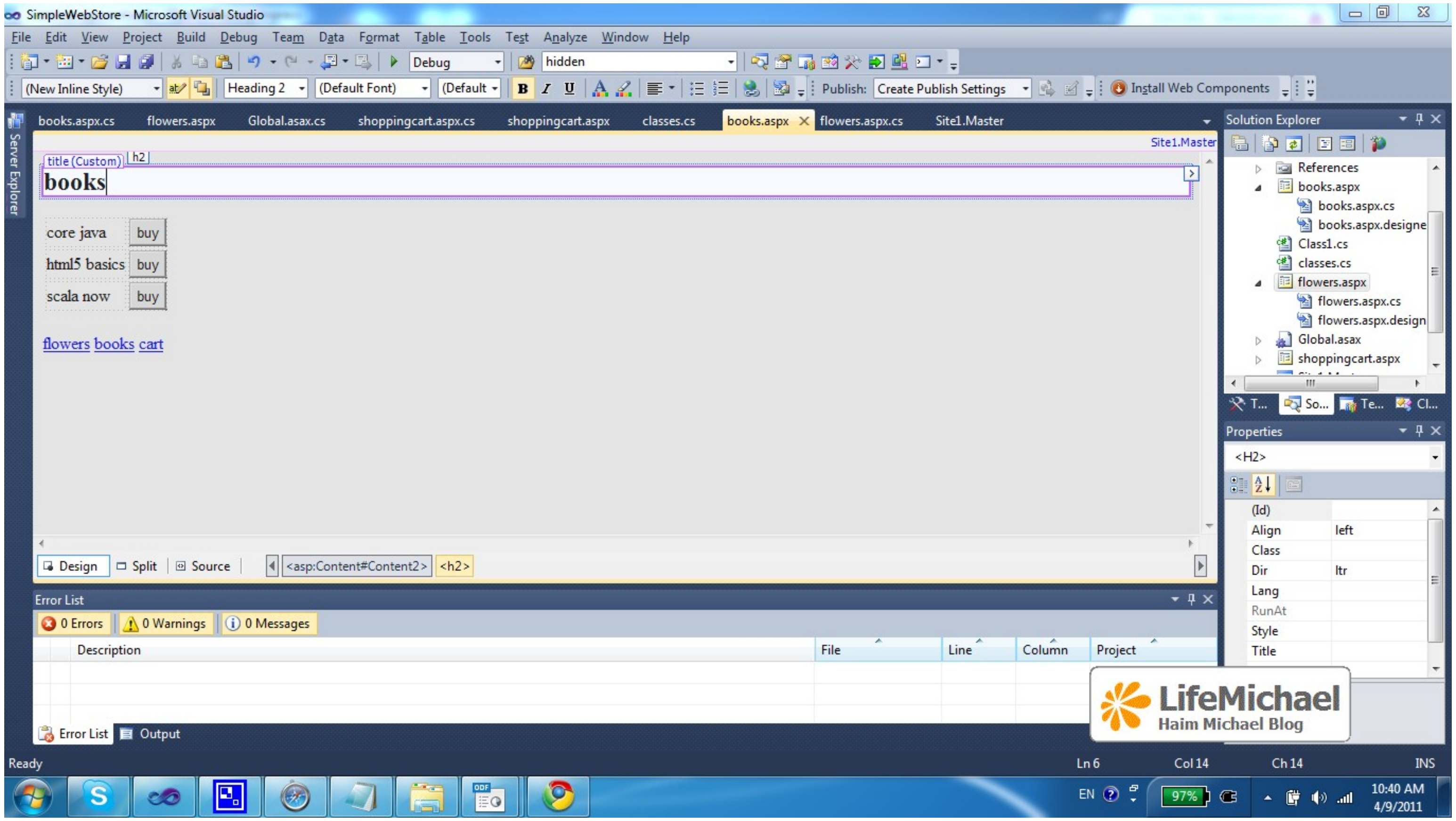
Sample

```
public class ShoppingCart
{
    public IDictionary<int, ShoppingCartLine> Lines =
        new Dictionary<int, ShoppingCartLine>();
    public void Add(Product product)
    {
        if (!Lines.Keys.Contains(product.Id))
        {
            Lines.Add(product.Id,new ShoppingCartLine(product,1));
        }
        else
        {
            ShoppingCartLine line = Lines[product.Id];
            line.Quantity += 1;
        }
    }
}
```

Sample

```
    public class ShoppingCartLine
    {
        public Product Product { get; set; }
        public int Quantity { get; set; }
        public ShoppingCartLine(Product prod, int quant)
        {
            Product = prod;
            Quantity = quant;
        }
    }
    public class Product
    {
        public string Name { get; set; }
        public double Price { get; set; }
        public int Id { get; set; }
        public Product(string nameVal, double priceVal, int idVal)
        {
            Name = nameVal;
            Price = priceVal;
            Id = idVal;
        }
    }
}
```

Sample


SimpleWebStore - Microsoft Visual Studio
File Edit View Project Build Debug Team Data Format Table Tools Test Analyze Window Help
Debug
hidden
(New Inline Style)
Heading 2
(Default Font)
Publish: Create Publish Settings
Install Web Components
books.aspx.cs flowers.aspx Global.asax.cs shoppingcart.aspx.cs shoppingcart.aspx classes.cs books.aspx flowers.aspx.cs Site1.Master
Solution Explorer
References
books.aspx
books.aspx.cs
books.aspx.designe
Class1.cs
classes.cs
flowers.aspx
flowers.aspx.cs
flowers.aspx.design
Global.asax
shoppingcart.aspx
Properties
<H2>
Error List
0 Errors 0 Warnings 0 Messages
Description File Line Column Project
Design Split Source
<asp:Content#Content2> <h2>
Ready Ln 6 Col 14 Ch 14 INS
LifeMichael
Haim Michael Blog

title (Custom) h2

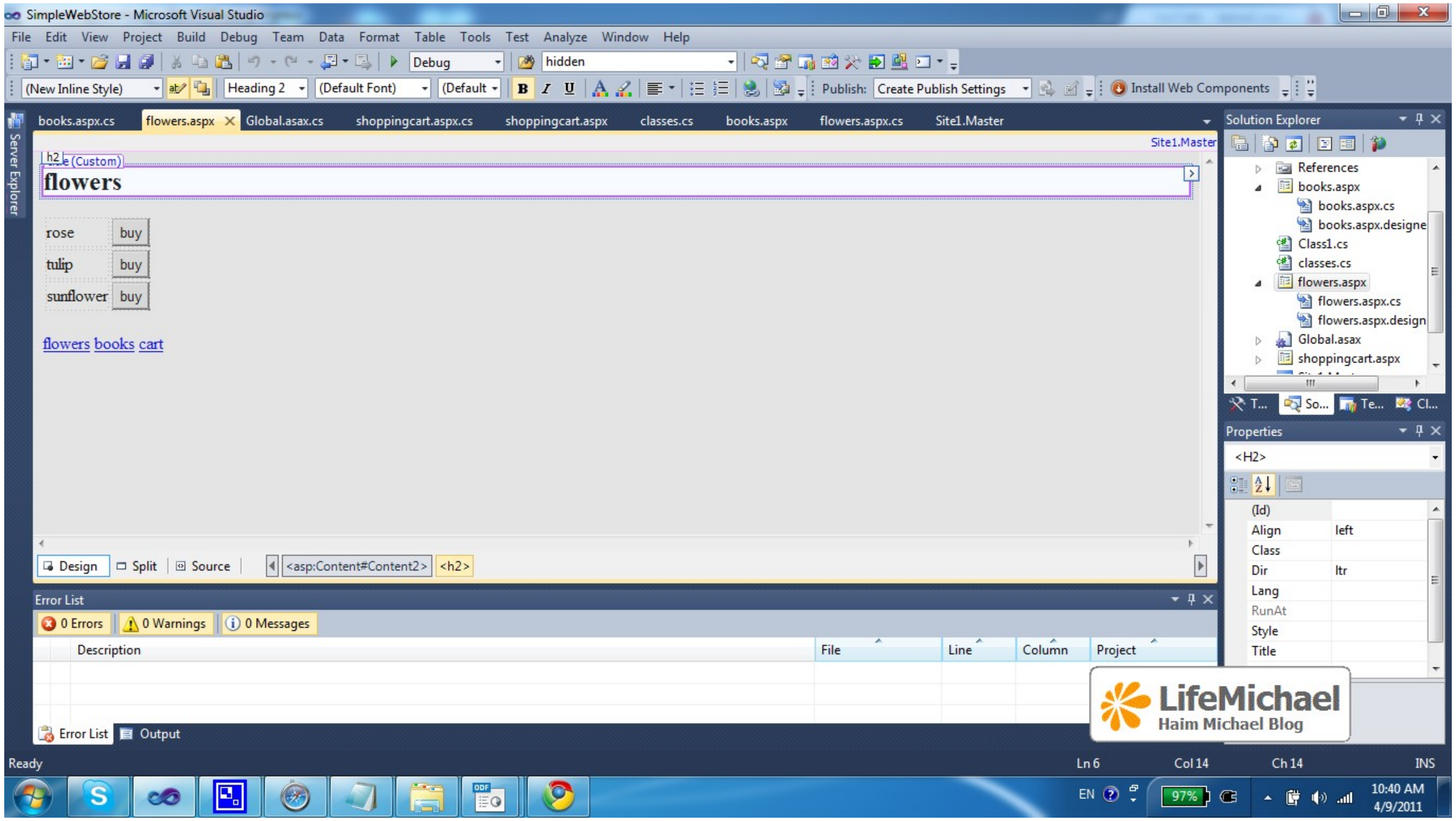
books

core java	buy
html5 basics	buy
scala now	buy

flowers books cart

(Id)	
Align	left
Class	
Dir	ltr
Lang	
RunAt	
Style	
Title	

Sample


SimpleWebStore - Microsoft Visual Studio
File Edit View Project Build Debug Team Data Format Table Tools Test Analyze Window Help
books.aspx.cs flowers.aspx Global.asax.cs shoppingcart.aspx.cs shoppingcart.aspx classes.cs books.aspx flowers.aspx.cs Site1.Master
flowers
rose buy
tulip buy
sunflower buy
flowers books cart
Design Split Source
Solution Explorer
References
books.aspx
books.aspx.cs
Class1.cs
classes.cs
flowers.aspx
flowers.aspx.cs
Global.asax
shoppingcart.aspx
Properties
<H2>
Align left
Dir ltr
Error List
0 Errors 0 Warnings 0 Messages
Description File Line Column Project
LifeMichael
Haim Michael Blog
Ready

Sample

The flowers.aspx and books.aspx Code Behind

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

namespace SimpleWebStore
{
    public partial class flowers : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
        protected void Button1_Click(object sender, EventArgs e)
        {
            ((ShoppingCart)Session["cart"]).Add(new Product("rose", 2.45, 1));
        }
        protected void Button2_Click(object sender, EventArgs e)
        {
            ((ShoppingCart)Session["cart"]).Add(new Product("tulip", 1.85, 2));
        }
```

Sample

```
        protected void Button3_Click(object sender, EventArgs e)
         {
            ((ShoppingCart)Session["cart"]).Add(new Product("sunflower", 4.45, 3));
         }

        protected void Button1_Click1(object sender, EventArgs e)
         {
            ((ShoppingCart)Session["cart"]).Add(new Product("core java", 42.45, 4));
         }

        protected void Button2_Click1(object sender, EventArgs e)
         {
            ((ShoppingCart)Session["cart"]).Add(new Product("html5 basics", 32.45, 5));
         }

        protected void Button3_Click1(object sender, EventArgs e)
         {
            ((ShoppingCart)Session["cart"]).Add(new Product("scala now", 72.25, 6));
         }
    }
}
```

Sample

The flowers.aspx Code

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/Site1.Master"
AutoEventWireup="true" CodeBehind="flowers.aspx.cs"
Inherits="SimpleWebStore.flowers" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
    flowers for sale
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="title" runat="server">
    <h2>flowers</h2>
</asp:Content>

<asp:Content ID="Content3" ContentPlaceHolderID="product1" runat="server">
    <tr><td><asp:Label ID="Label1" runat="server" Text="rose" /></td>
    <td><asp:Button ID="Button1" runat="server" Text="buy"
    onclick="Button1_Click" /></td></tr>
</asp:Content>
```

Sample

```
<asp:Content ID="Content4" ContentPlaceHolderID="product2" runat="server">
    <tr><td><asp:Label ID="Label2" runat="server" Text="tulip" /></td>
    <td><asp:Button ID="Button2" runat="server" Text="buy"
    onclick="Button2_Click" /></td></tr>
</asp:Content>

<asp:Content ID="Content5" ContentPlaceHolderID="product3" runat="server">
    <tr><td><asp:Label ID="Label3" runat="server" Text="sunflower" /></td>
    <td><asp:Button ID="Button3" runat="server" Text="buy"
    onclick="Button3_Click" /></td></tr>
</asp:Content>
```

Sample

The books.aspx Code

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master"
AutoEventWireup="true" CodeBehind="flowers.aspx.cs"
Inherits="SimpleWebStore.flowers" %>

<asp:Content ID="Content1" ContentPlaceHolderID="head" runat="server">
    flowers for sale
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="title" runat="server">
    <h2>books</h2>
</asp:Content>
```

Sample

```
<asp:Content ID="Content3" ContentPlaceHolderID="product1" runat="server">
    <tr><td><asp:Label ID="Label1" runat="server" Text="core java" /></td>
    <td><asp:Button ID="Button1" runat="server" Text="buy"
    onclick="Button1_Click1" /></td></tr>
</asp:Content>

<asp:Content ID="Content4" ContentPlaceHolderID="product2" runat="server">
    <tr><td><asp:Label ID="Label2" runat="server" Text="html5 basics" /></td>
    <td><asp:Button ID="Button2" runat="server" Text="buy"
    onclick="Button2_Click1" /></td></tr>
</asp:Content>

<asp:Content ID="Content5" ContentPlaceHolderID="product3" runat="server">
    <tr><td><asp:Label ID="Label3" runat="server" Text="scala now" /></td>
    <td><asp:Button ID="Button3" runat="server" Text="buy"
    onclick="Button3_Click1" /></td></tr>
</asp:Content>
```

Sample

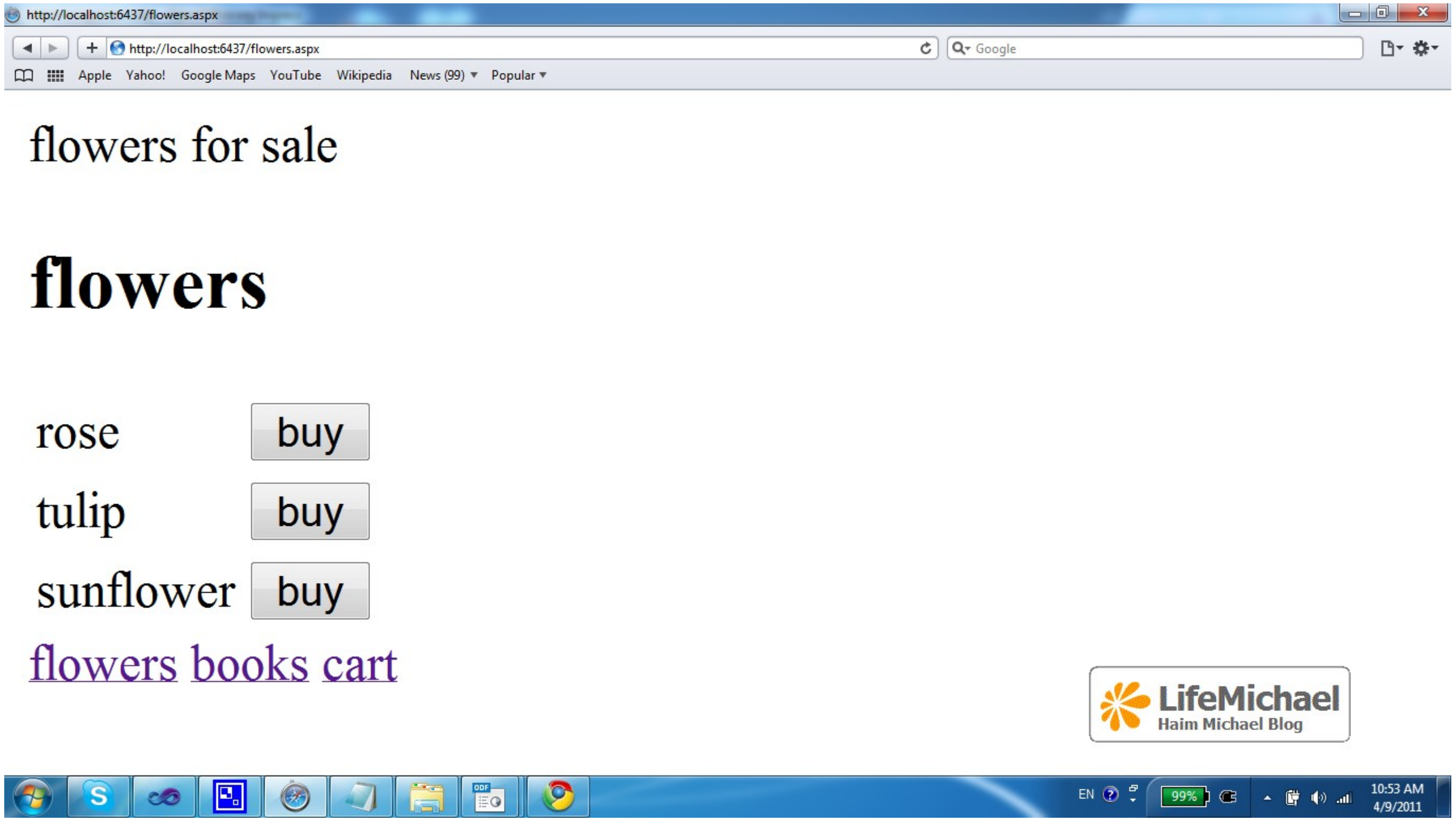
The Site1.Master Code

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="Site1.master.cs"
Inherits="SimpleWebStore.Site1" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

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

Sample

```
<body>
    <form id="form1" runat="server">
    <div>
        <asp:ContentPlaceHolder ID="title" runat="server">
        </asp:ContentPlaceHolder>
        <table>
        <asp:ContentPlaceHolder ID="product1" runat="server">
        </asp:ContentPlaceHolder>
        <asp:ContentPlaceHolder ID="product2" runat="server">
        </asp:ContentPlaceHolder>
        <asp:ContentPlaceHolder ID="product3" runat="server">
        </asp:ContentPlaceHolder>
        </table>
    </div>
    </form>
    <a href="flowers.aspx">flowers</a>
    <a href="books.aspx">books</a>
    <a href="shoppingcart.aspx">cart</a>
</body>
</html>
```

Sample



http://localhost:6437/flowers.aspx

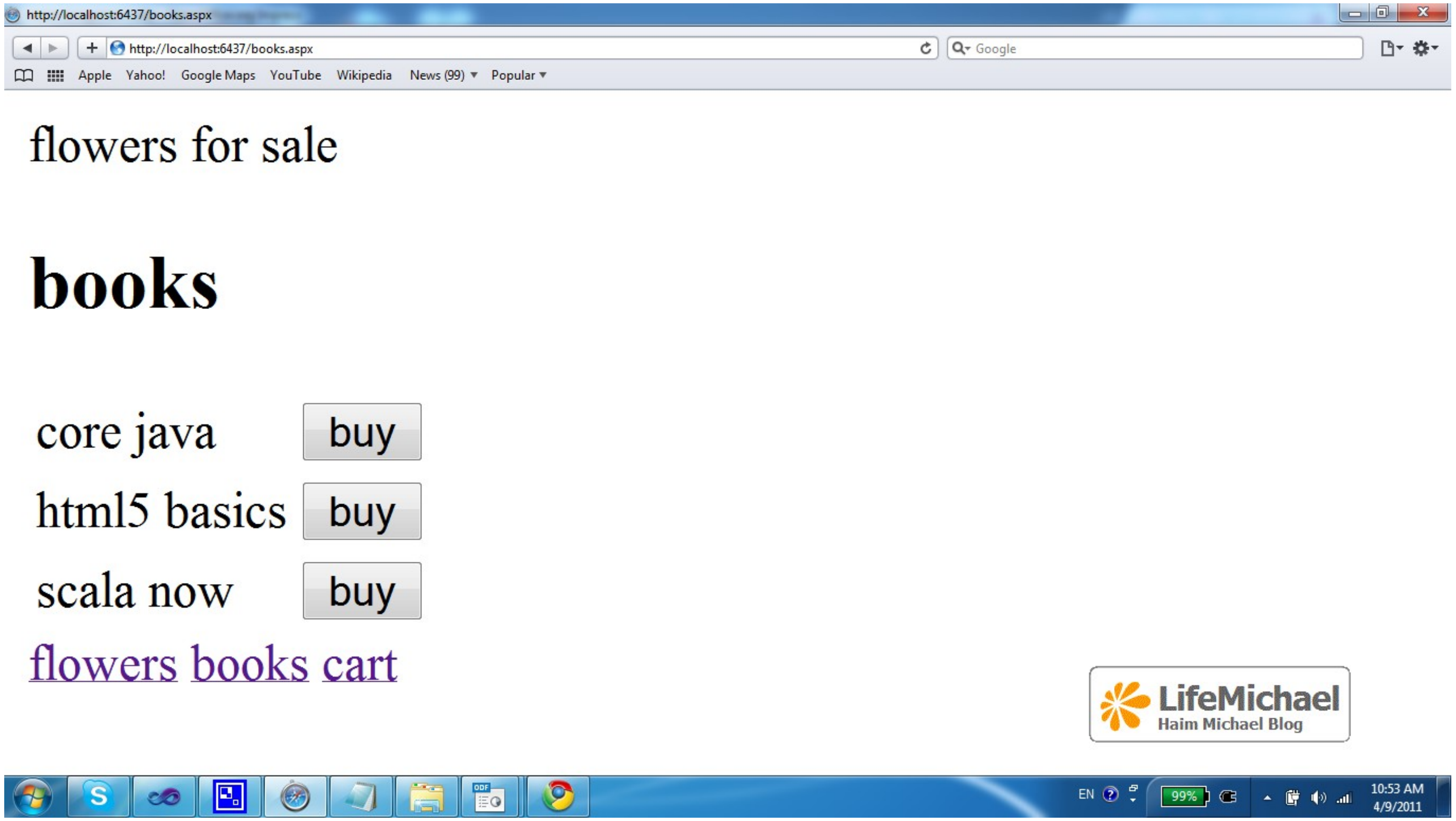
flowers for sale

flowers

rose buy
tulip buy
sunflower buy

flowers books cart

Sample


http://localhost:6437/books.aspx

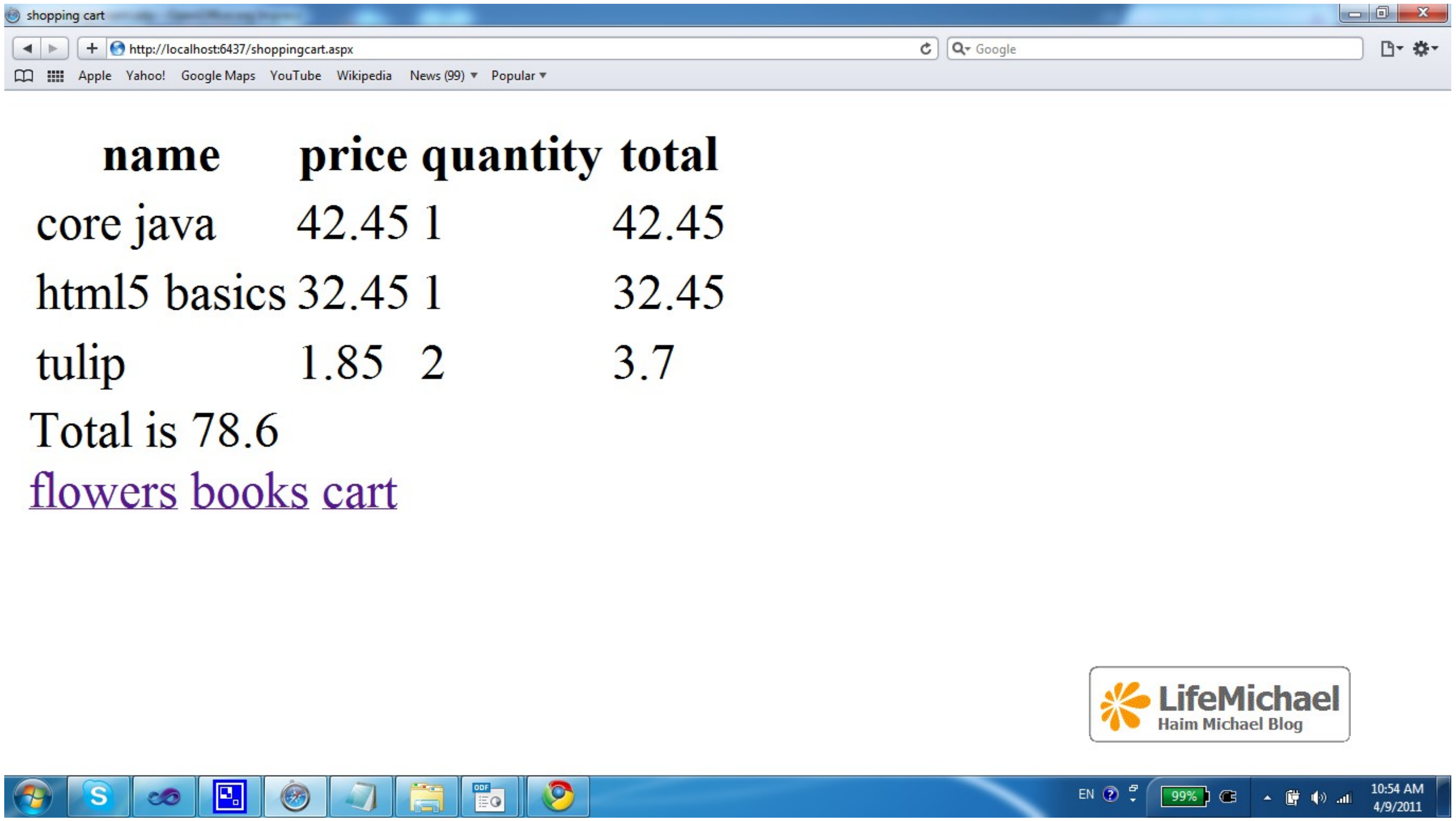
flowers for sale

books

core java buy
html5 basics buy
scala now buy

flowers books cart

Sample



shopping cart

http://localhost:6437/shoppingcart.aspx

name	price	quantity	total
core java	42.45	1	42.45
html5 basics	32.45	1	32.45
tulip	1.85	2	3.7

Total is 78.6

flowers books cart