

Response

Emitting HTML Content

- We can easily emit HTML content back to the client by calling the `Write` method on the `HttpResponse` object we are working with.

Emitting HTML Content

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
Inherits="WebApplication7.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 runat="server">
    <asp:Button ID="Button1" runat="server" Text="press here"
      onclick="Button1_Click" />
  </form>
</body>
</html>
```



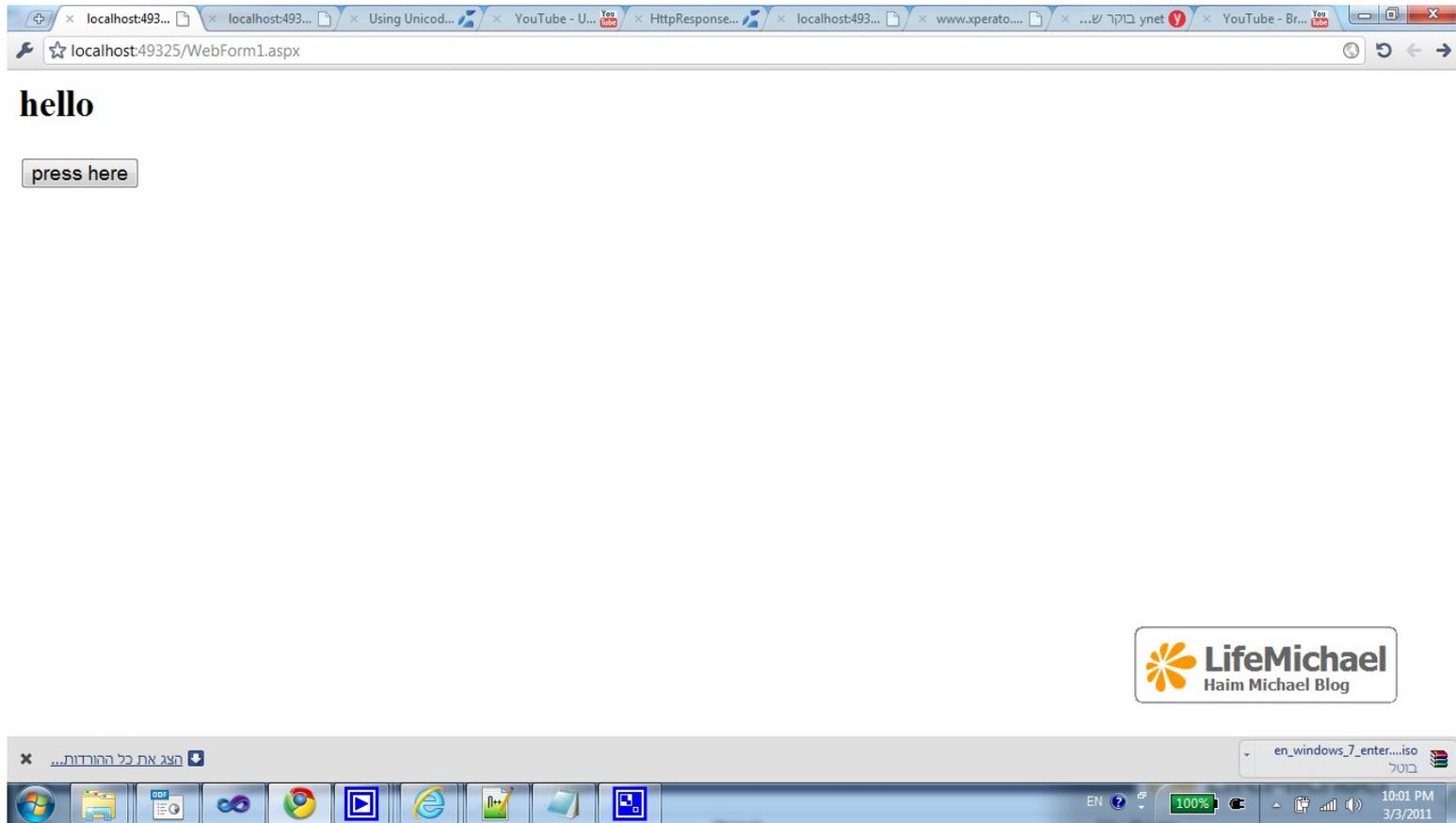
Emitting HTML Content

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

namespace WebApplication7
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.Write("<h2>hello</h2>");
        }
    }
}
```

Emitting HTML Content



Emitting File Content

- The `Response.WriteFile` method allows us to emit the content of a file, located on the server side within the root directory of the website, back to the web browser.

Emitting File Content

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="WebApplication7.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 runat="server">
        <asp:Button ID="Button1" runat="server" Text="press here"
            onclick="Button1_Click" />
    </form>
</body>
</html>
```



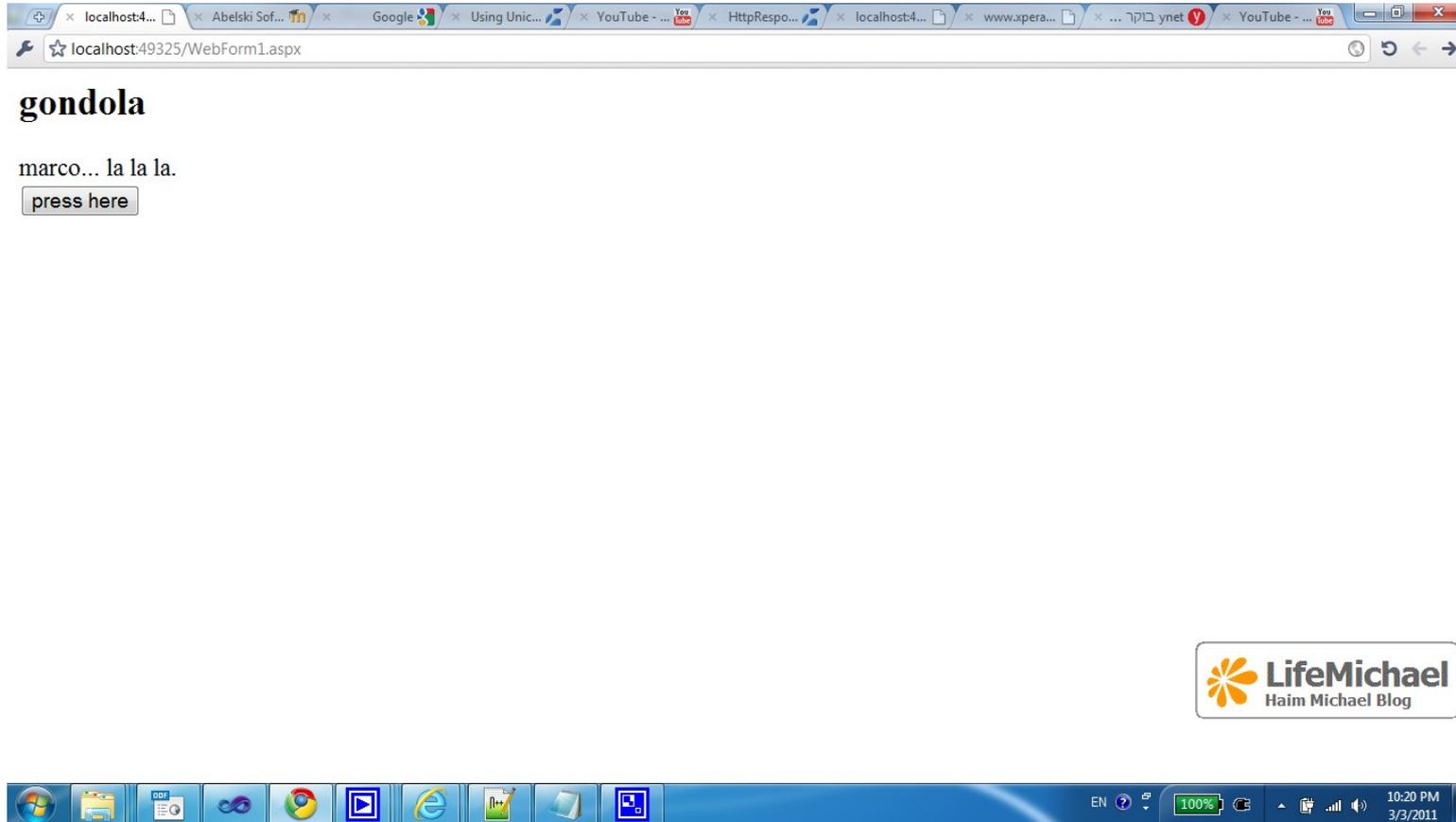
Emitting File Content

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

namespace WebApplication7
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.WriteFile("detailsnice.txt");
        }
    }
}
```

Emitting File Content



The ContentEncoding Property

- The `Response.ContentEncoding` property allows us to set the HTTP character set of the output stream.

The ContentEncoding Property

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="WebApplication7.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 runat="server">
        <asp:Button ID="Button1" runat="server" Text="press here"
            onclick="Button1_Click" />
    </form>
</body>
</html>
```

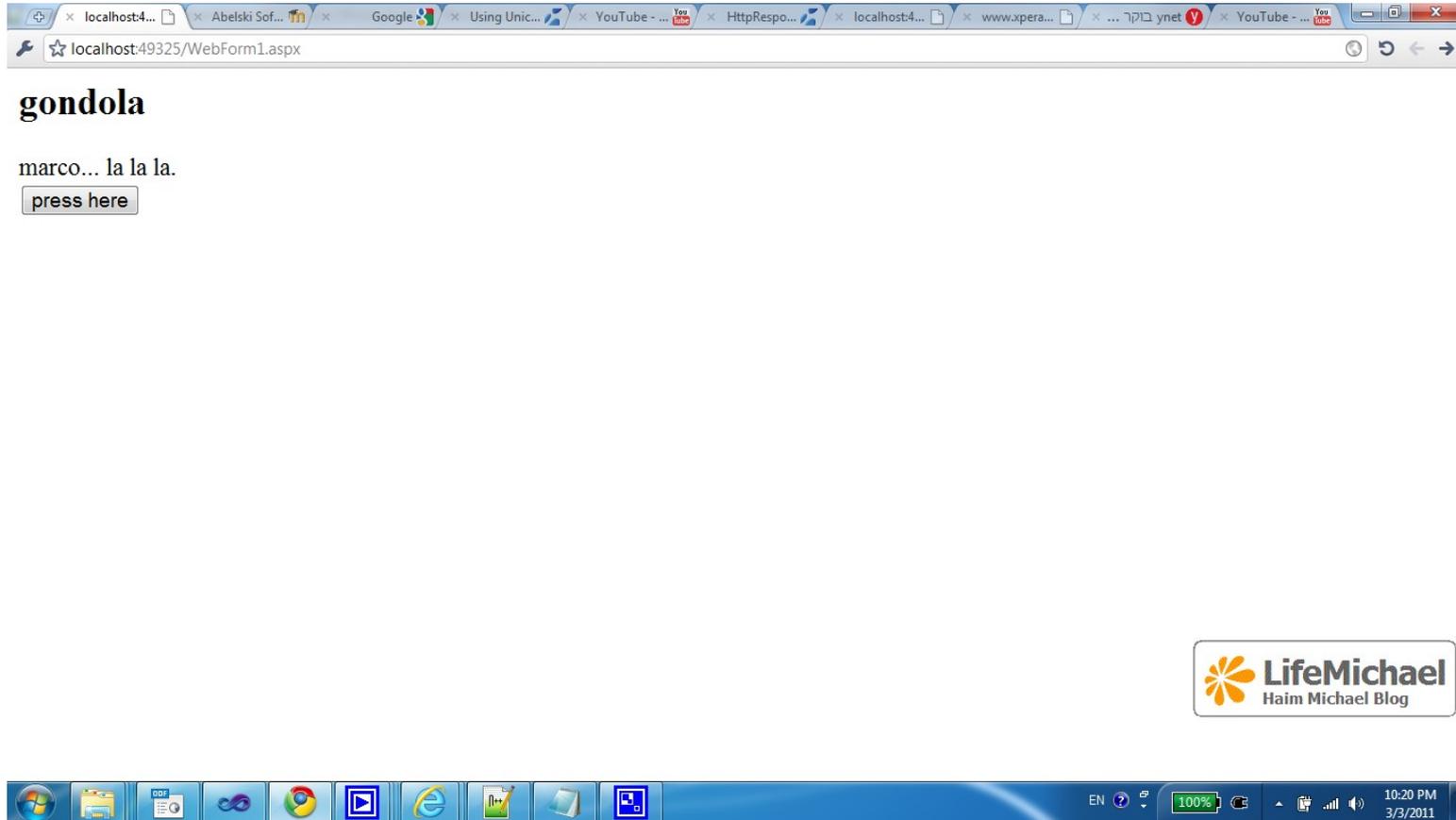


The ContentEncoding Property

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

namespace WebApplication7
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.ContentEncoding = Encoding.UTF8;
            Response.WriteFile("detailsnice.txt");
        }
    }
}
```

The ContentEncoding Property



Redirecting Users

- The response can include the required HTML code for redirecting the user to another URL address.
- We can call the `Redirect` method on `HttpResponse` object in order to redirect him to another URL address.

Redirecting Users

```
<%@ Page Language="C#" AutoEventWireup="true"
    CodeBehind="WebForm1.aspx.cs" Inherits="WebApplication7.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 runat="server">
        <asp:Button ID="Button1" runat="server" Text="press here"
            onclick="Button1_Click" />
    </form>
</body>
</html>
```



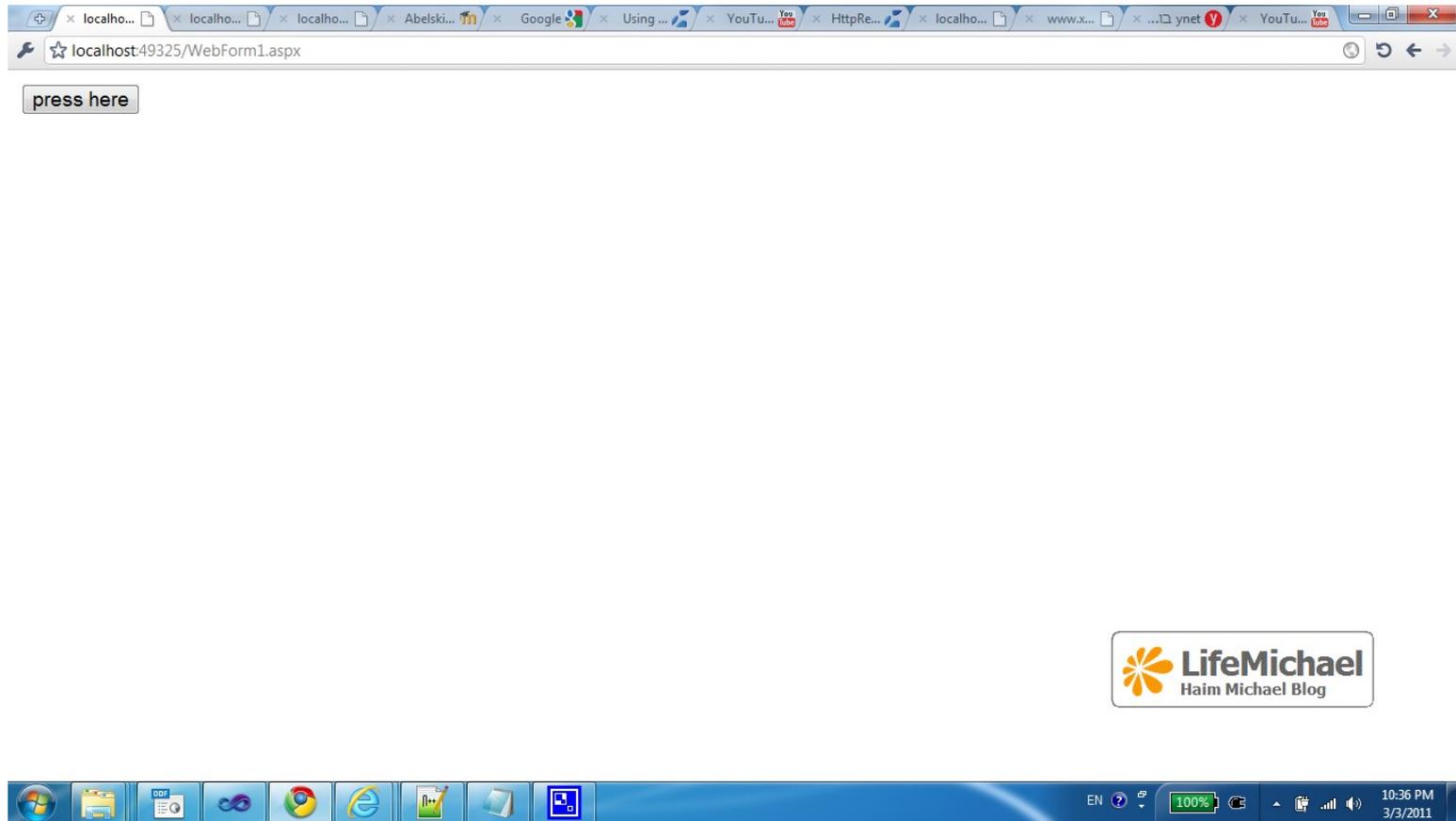
Redirecting Users

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

namespace WebApplication7
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }

        protected void Button1_Click(object sender, EventArgs e)
        {
            Response.Redirect("http://www.xperato.com");
        }
    }
}
```

Redirecting Users



Redirecting Users

The screenshot shows a Joomla! website for XPerato. The browser's address bar displays www.xperato.com/en/joomla/. The website header includes the XPerato logo and the text "Software Development Training Services". A navigation menu contains links for Home, Courses, Services, Articles, The News, About Us, and Contact Us, along with a search box. The main content area features an article titled "The Passion for Teaching!" with a sub-heading "The jQuery Mobile Framework". The article is dated Tuesday, 15 February 2011 02:31 and is written by Haim Michael. The text describes jQueryMobile as a JavaScript framework for developing cross-platform, touch-based user interfaces for mobile devices. A jQuery logo is visible next to the text. The right sidebar contains a poll titled "Favorite way to learn?" with options: Training Course, Reading Books, Free Web Resources, Private Teacher, and All of the Above! Below the poll are sections for "Who's Online" (1 guest online) and "Advertisement" with featured links to Abelski Free Courses, LifeMichael Haim Michael Blog, and Holon Institute of Technology. The Windows taskbar at the bottom shows the system clock as 10:37 PM on 3/3/2011.

Transferring Users

- In order to transfer the user to another *.aspx file in the same virtual directory we should call the `Transfer()` method on the `HttpServerUtility` object we can access by referring the `Server` property.

Transferring Users

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

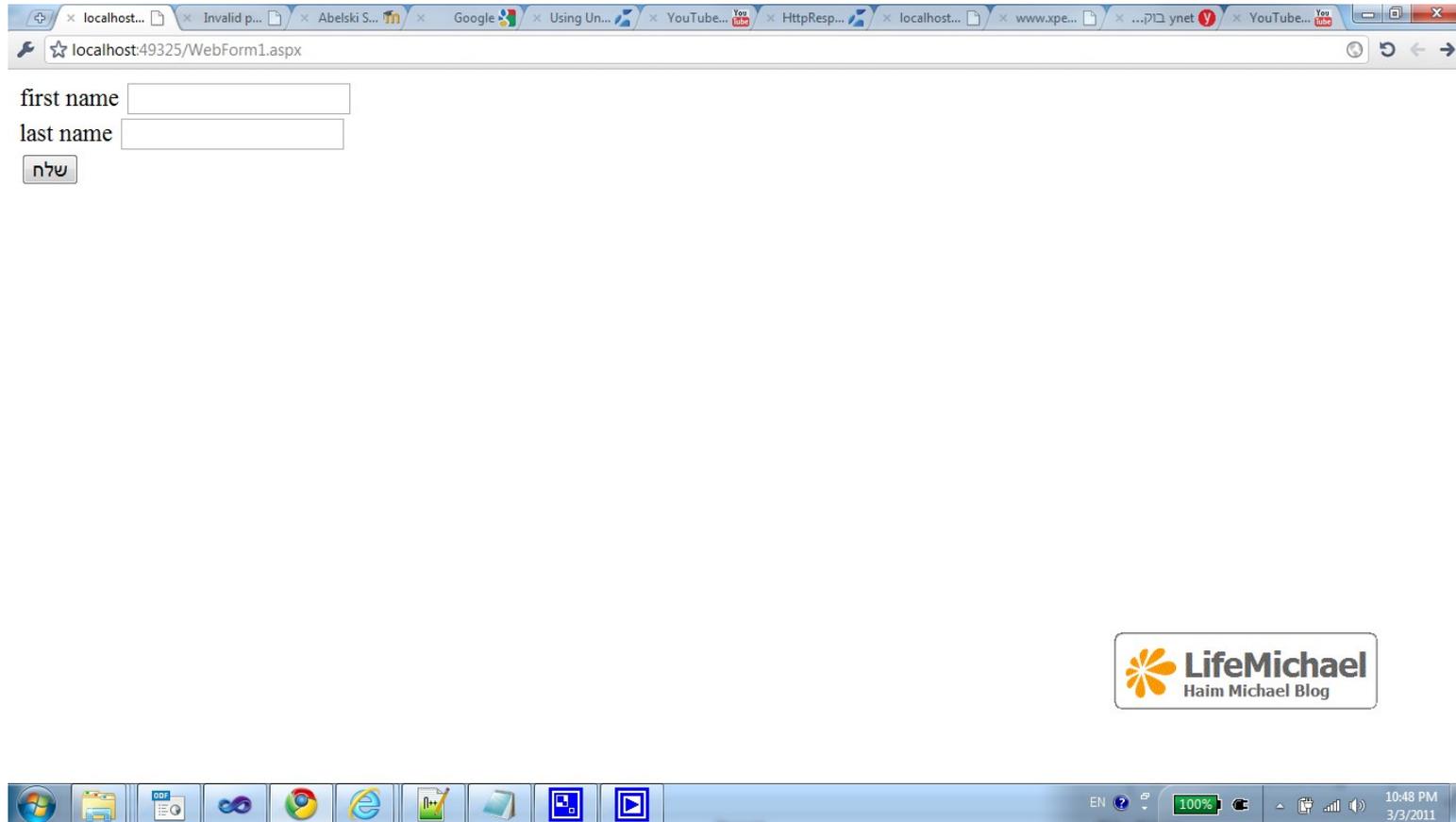
namespace WebApplication7
{
    public partial class WebForm1 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
        }
        protected void Button1_Click(object sender, EventArgs e)
        {
            Server.Transfer("HTMLPage1.htm");
        }
    }
}
```



Transferring Users

```
<!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>
  <title></title>
</head>
<body>
  <form method="post" action="WebForm1.aspx">
    first name <input type="text" name="firstName" />
    <br />
    last name <input type="text" name="lastName" />
    <br />
    <input type="submit" />
  </form>
</body>
</html>
```

Transferring Users



Cookies

- We can indirectly add the required HTTP headers to our response in order to add a cookie in the browser that receives it.
- The simplest way to do it involves with instantiating the `HttpCookie` class and adding the new created object to the cookies collection we can access through the `Cookies` property defined in `HttpResponse`.

Cookies

```
namespace WebApplication7
{
    public partial class WebForm2 : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (Request.Cookies.Keys.Count == 0)
            {
                Response.Write("<br>no cookies</br>");
                HttpCookie cookie = new HttpCookie("user", "123123");
                Response.Cookies.Add(cookie);
            }
        }
    }
}
```



Cookies

```
else
{
    if (Request.Cookies.Keys.Count == 1)
    {
        Response.Write("<h2>there is one cookie</h2>");
    }
    else
    {
        Response.Write("<h2>there are "
            + Request.Cookies.Keys.Count + " cookies</h2>");
    }
}
}
}
```

Cookies



there is one cookie

