Master Pages

Introduction

- Master pages define a common UI layout shared by all pages or a sub set of pages in our web site.
- The `*.master` file defines kind of placeholder areas at which other `*.aspx` pages can plug. When this is the case, the `*.aspx` pages are known as content pages.

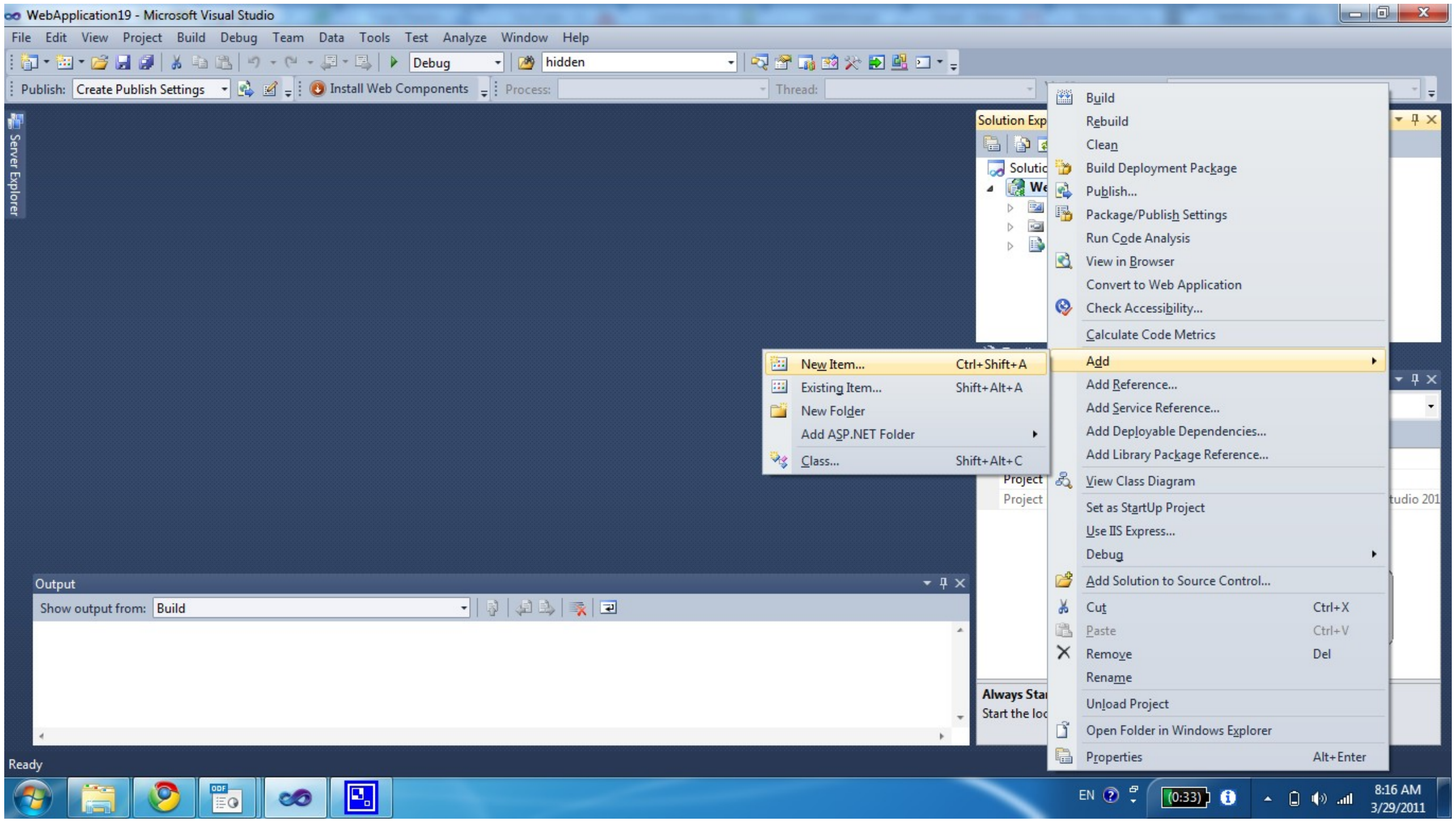
The Content Pages

- The `*.aspx` pages we plug into the master page are known as content pages and the code they include is a bit different comparing with any other simple `*.aspx` page we usually code.
- The *.aspx pages don't include the HTML `<form>` elements. The master page already includes it.
- When the user request a `*.aspx` page that refers a master page he will get the master page with the requested `*.aspx` embedded within.

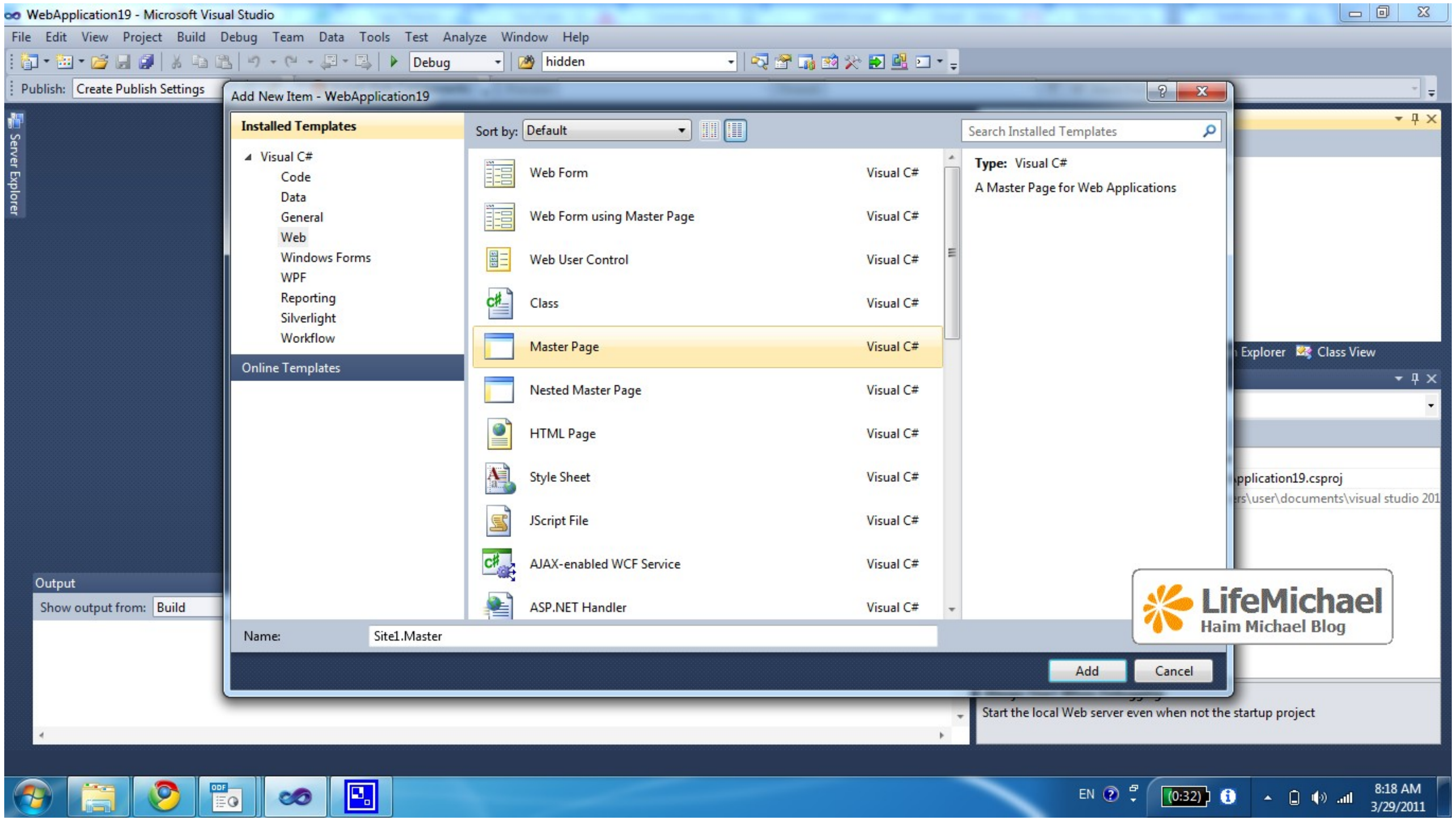
Creating Master Page

- In order to create a new master page we just need to right click our project, select 'Add Item' and select the 'Master Page' option.

Creating Master Page



Creating Master Page



Web Form	Visual C#
Web Form using Master Page	Visual C#
Web User Control	Visual C#
Class	Visual C#
Master Page	Visual C#
Nested Master Page	Visual C#
HTML Page	Visual C#
Style Sheet	Visual C#
JScript File	Visual C#
AJAX-enabled WCF Service	Visual C#
ASP.NET Handler	Visual C#

Type: Visual C#

A Master Page for Web Applications

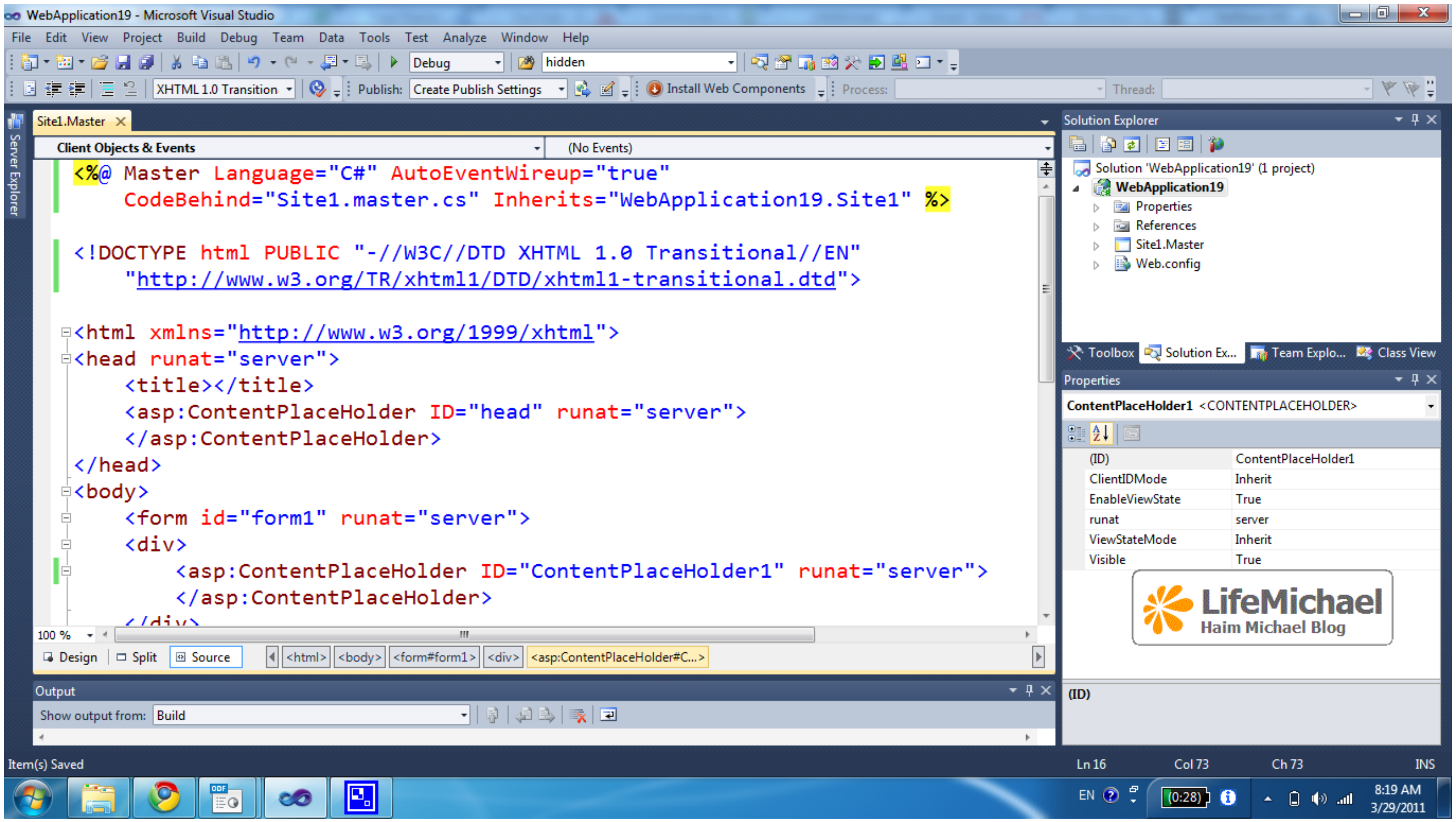
Name: Site1.Master

pplication19.csproj

rs\user\documents\visual studio 201

Start the local Web server even when not the startup project

Creating Master Page



```
<%@ Master Language="C#" AutoEventWireup="true"
    CodeBehind="Site1.master.cs" Inherits="WebApplication19.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>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:ContentPlaceHolder ID="ContentPlaceHolder1" runat="server">
        </asp:ContentPlaceHolder>
    </div>
```

Properties

ContentPlaceHolder1 <CONTENTPLACEHOLDER>

(ID)	ContentPlaceHolder1
ClientIDMode	Inherit
EnableViewState	True
runat	server
ViewStateMode	Inherit
Visible	True

The `MasterPage` Class

- When creating a master page the code behind includes the definition of a class that extends the `MasterPage` class.
- This class belongs to the `System.Web.UI` namespace.

The `MasterPage` Class

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

namespace WebApplication19
{
    public partial class Site1 : System.Web.UI.MasterPage
    {
        protected void Page_Load(object sender, EventArgs e)
        {

        }
    }
}
```

The `*.Master` File

- The place holders are marked using the `<asp:ContentPlaceHolder>` elements.
- We can place within elements of this type content that will apply when the `*.aspx` file doesn't specify a specific content.
- The master page can define as many content placeholders as needed.
- Each single master page can nest additional master pages within it.

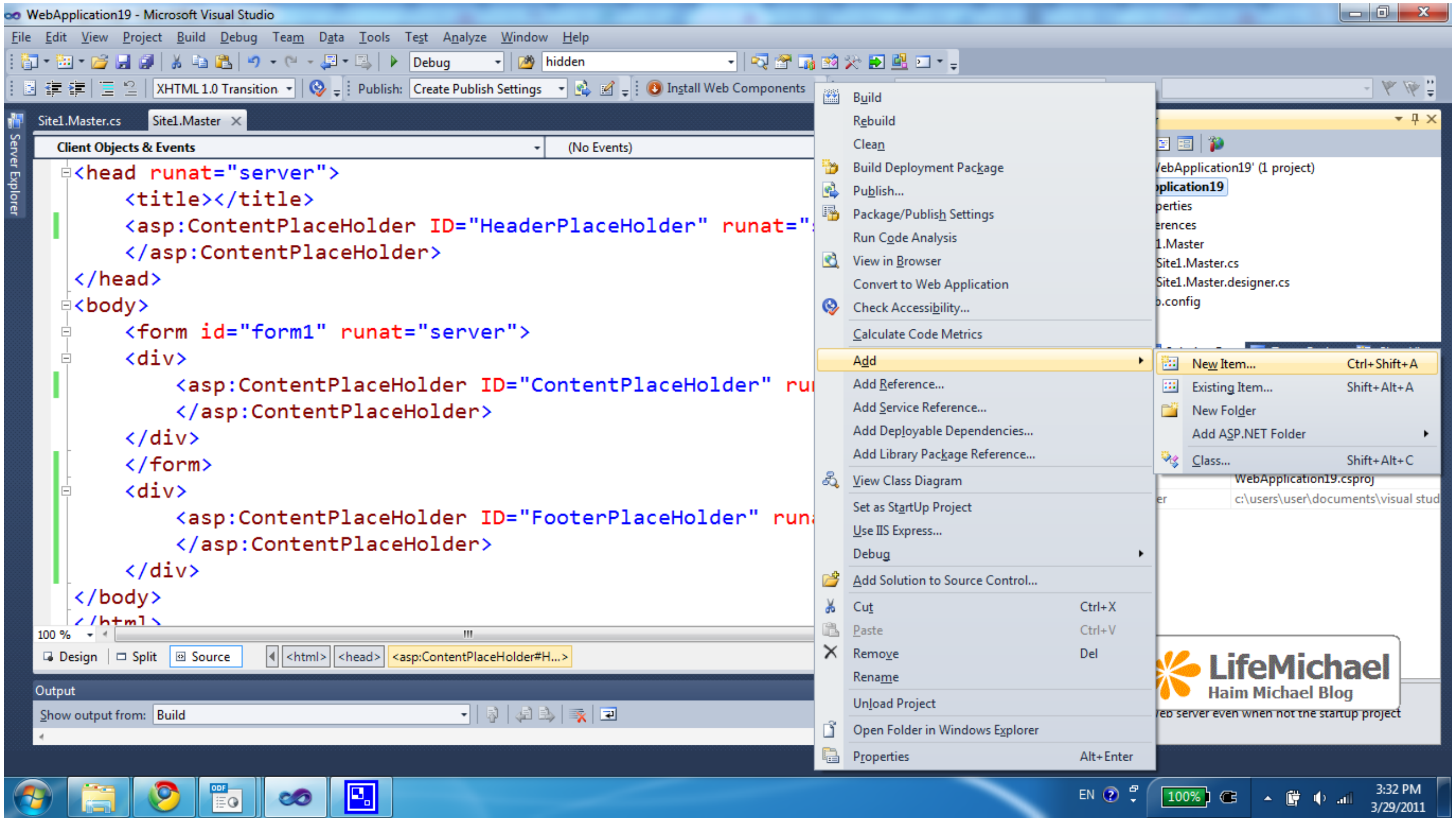
The *.Master File

```
<%@ Master Language="C#" AutoEventWireup="true"
    CodeBehind="Site1.master.cs" Inherits="WebApplication19.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>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:ContentPlaceHolder ID="ContentPlaceHolder1"
           runat="server">
        </asp:ContentPlaceHolder>
    </div>
    </form>
</body>
</html>
```

The *.aspx File

- We can easily walk through a wizard for adding a new web form that uses a master page our web project includes.

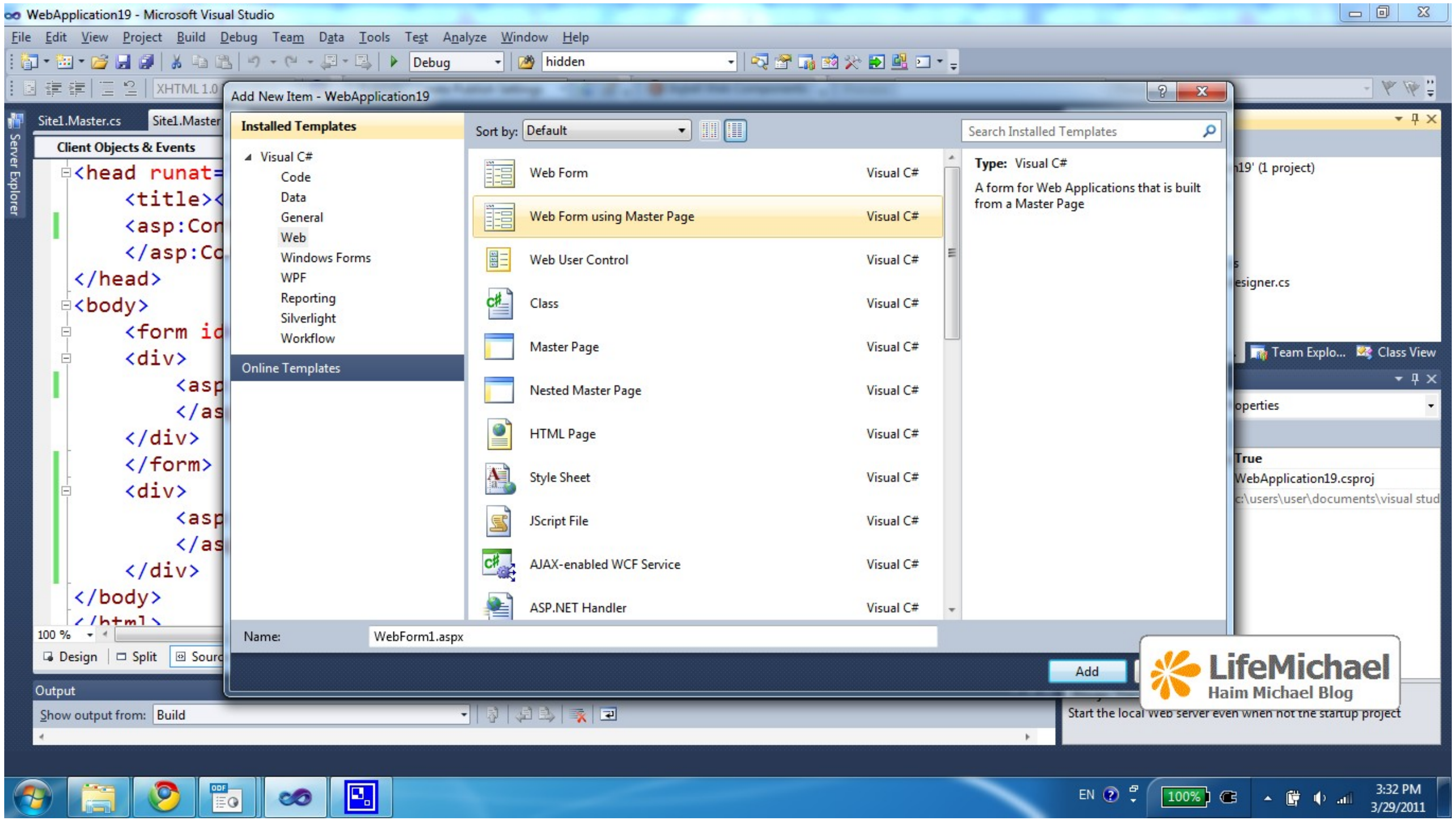
The *.aspx File



```
<head runat="server">
    <title></title>
    <asp:ContentPlaceHolder ID="HeaderPlaceHolder" runat="
    </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:ContentPlaceHolder ID="ContentPlaceHolder" ru
        </asp:ContentPlaceHolder>
    </div>
    </form>
    <div>
        <asp:ContentPlaceHolder ID="FooterPlaceHolder" run
        </asp:ContentPlaceHolder>
    </div>
</body>
```

(c) 2010 Haim Michael. All Rights Reserved.

The *.aspx File



```
<head runat=
    <title><
    <asp:Cor
    </asp:Co
</head>
<body>
    <form id
    <div>
        <asp
        </as
    </div>
    </form>
    <div>
        <asp
        </as
    </div>
</body>
</html>
```

Add New Item - WebApplication19

Installed Templates

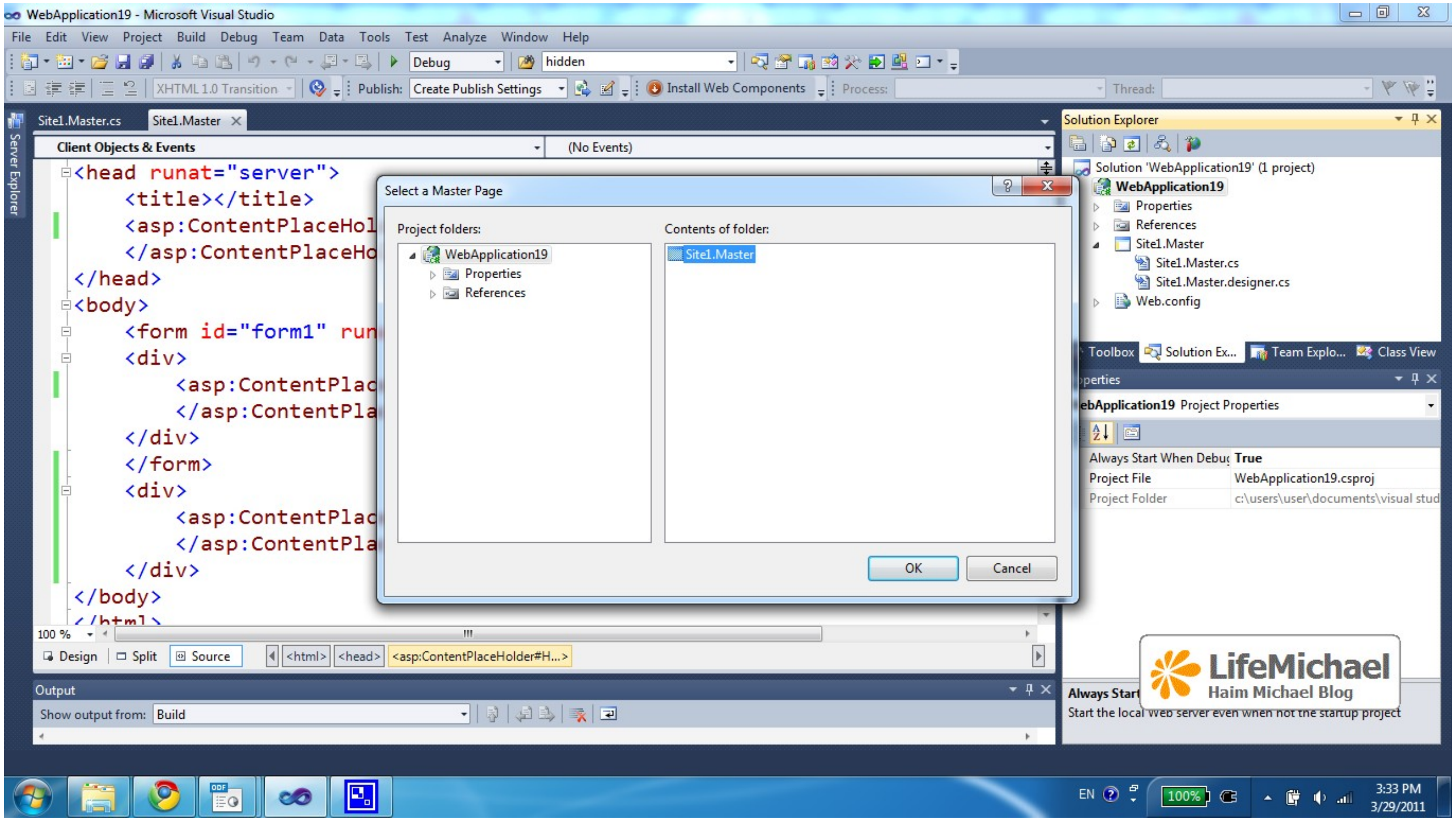
Template	Type
Web Form	Visual C#
Web Form using Master Page	Visual C#
Web User Control	Visual C#
Class	Visual C#
Master Page	Visual C#
Nested Master Page	Visual C#
HTML Page	Visual C#
Style Sheet	Visual C#
JScript File	Visual C#
AJAX-enabled WCF Service	Visual C#
ASP.NET Handler	Visual C#

Type: Visual C#

A form for Web Applications that is built from a Master Page

Name: WebForm1.aspx

The *.aspx File



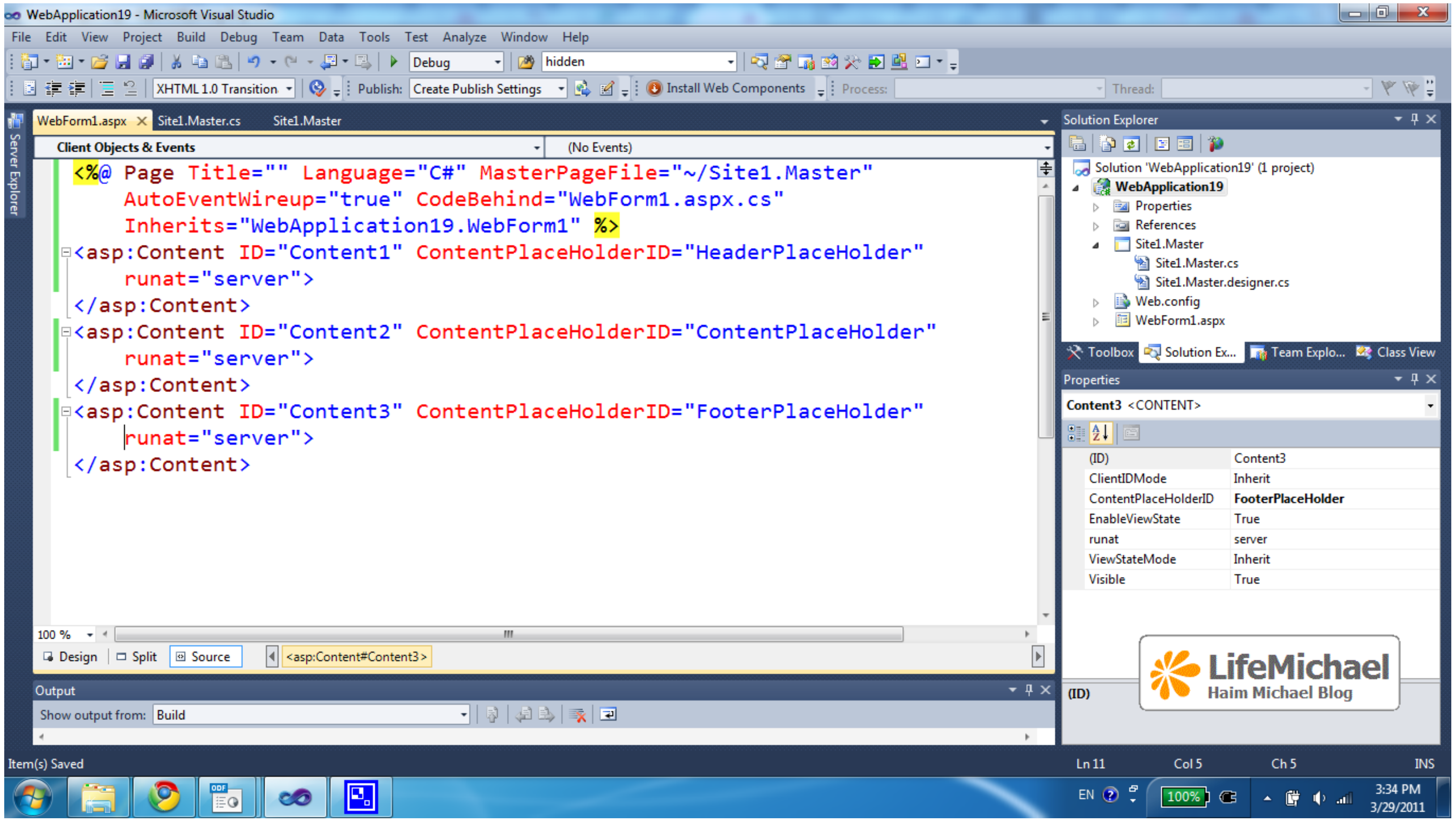
```
<head runat="server">
    <title></title>
    <asp:ContentPlaceHol
    </asp:ContentPlaceHo
</head>
<body>
    <form id="form1" run
    <div>
        <asp:ContentPlac
        </asp:ContentPla
    </div>
    </form>
    <div>
        <asp:ContentPlac
        </asp:ContentPla
    </div>
</body>
</html>
```

Always Start When Debuç	True
Project File	WebApplication19.csproj
Project Folder	c:\users\user\documents\visual stud

Show output from: Build

(c) 2010 Haim Michael. All Rights Reserved.

The `*.aspx` File



```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master"
    AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
    Inherits="WebApplication19.WebForm1" %>
<asp:Content ID="Content1" ContentPlaceHolderID="HeaderPlaceHolder"
    runat="server">
</asp:Content>
<asp:Content ID="Content2" ContentPlaceHolderID="ContentPlaceHolder"
    runat="server">
</asp:Content>
<asp:Content ID="Content3" ContentPlaceHolderID="FooterPlaceHolder"
    runat="server">
</asp:Content>
```

Solution 'WebApplication19' (1 project)
- WebApplication19
  - Properties
  - References
  - Site1.Master
    - Site1.Master.cs
    - Site1.Master.designer.cs
  - Web.config
  - WebForm1.aspx

Properties

Content3 <CONTENT>

(ID)	Content3
ClientIDMode	Inherit
ContentPlaceHolderID	FooterPlaceHolder
EnableViewState	True
runat	server
ViewStateMode	Inherit
Visible	True

The `*.aspx` File

- The auto generated `*.aspx` file we get when following this wizard includes `asp:Content` elements for each one of the place holders the master page includes.
- The content page `*.aspx` file doesn't define the page. The outer shell is already provided by the master page.
- The content page `*.aspx` file is relatively clean. It doesn't need to include any of the details defined in the master page.

Sample

Site1.Master

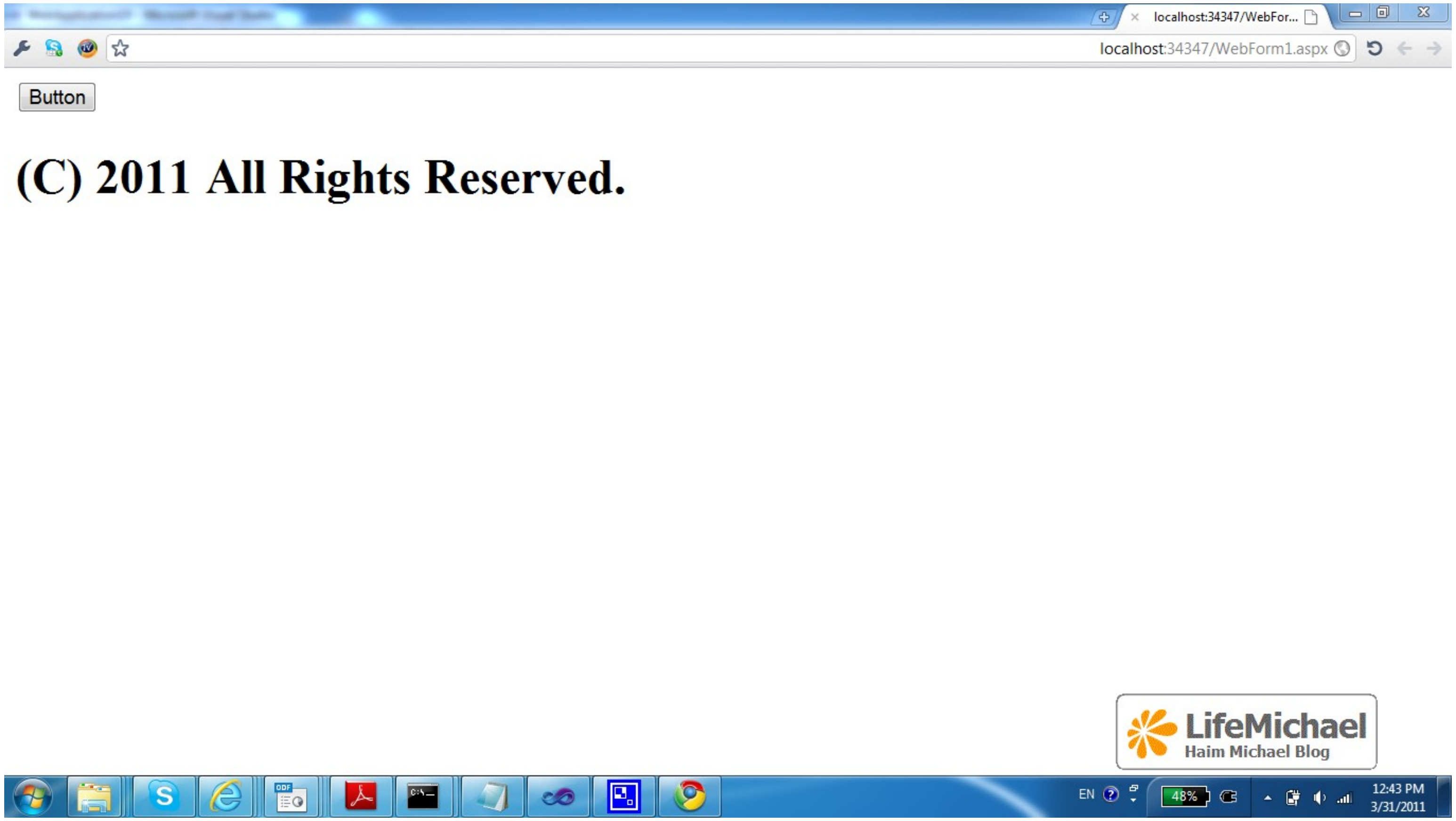
```
<%@ Master Language="C#" AutoEventWireup="true"
    CodeBehind="Site1.master.cs" Inherits="WebApplication19.Site1" %>

<!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="HeaderPlaceHolder" runat="server">
    </asp:ContentPlaceHolder>
</head>
<body>
    <form id="form1" runat="server"><div>
        <asp:ContentPlaceHolder ID="MainContentPlaceHolder" runat="server">
        </asp:ContentPlaceHolder>
    </div></form>
    <div>
        <h1>
            <asp:ContentPlaceHolder ID="FooterPlaceHolder" runat="server">
            </asp:ContentPlaceHolder>
        </h1>
    </div>
</body>
</html>
```

Sample

WebForm1.aspx

```
<%@ Page Title="" Language="C#" MasterPageFile="~/Site1.Master"
    AutoEventWireup="true" CodeBehind="WebForm1.aspx.cs"
    Inherits="WebApplication19.WebForm1" %>

<asp:Content ID="Content1" ContentPlaceHolderID="HeaderPlaceHolder" runat="server">
    <script language="javascript" type="text/javascript">
        function sum(num1, num2) {return num1 + num2;}
    </script>
</asp:Content>

<asp:Content ID="Content2" ContentPlaceHolderID="MainContentPlaceHolder"
    runat="server">
    <asp:Button ID="OK" runat="server" Text="Button" />
</asp:Content>

<asp:Content ID="Content3" ContentPlaceHolderID="FooterPlaceHolder" runat="server">
    (C) 2011 All Rights Reserved.
</asp:Content>
```

Sample



(C) 2011 All Rights Reserved.

Pages Life Cycle

- When programmatically introducing changes into our master and content pages it is important to know understand the the order in which the respective events fire.
- The ASP.NET server side starts with creating the master page controls and then it continues with creating the child controls the content page holds.
- Once the controls were created the Page.Init event for the master page is fired. The Page.Init event for the content page is fired right afterwards.

Pages Life Cycle

- The same happens with the rest of the events. First the Page.Load event is fired for the master page. The Page.Load event is fired for the content page afterwards.
- Therefore, when there is a conflict, customization we introduce into the content page will take precedence over customization that was introduced in the same stage in the master page.

Default Content

- When we place a `ContentPlaceHolder` in a master page we can include default content. That default content will be used wen the content page doesn't supply a corresponding content.

```
<asp:ContentPlaceHolder id="title_content" runat="server">
Simple Title
</asp:ContentPlaceHolder>
```

- In order to see the default content that was set in the master page we should delete the `<Content>` tag from the content page. Otherwise, it will still override the default content.

Tables and CSS

- We can place in our master page a table (in HTML) and place the `ContentPlaceHolder` elements within the cells in order to allow the content page that uses our master page to specify the exact content in each one of the place holders.

Nesting Master Pages

- We can nest a master page within another master page. The first refers the second as its master.
- We can use this capability for setting an overall master page that sets the look of the whole site allowing different parts to add their own child master pages.

Nesting Master Pages

```
<%@ Master Language="C#" AutoEventWireup="true"
CodeBehind="Site1.master.cs" Inherits="WebApplication20.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>
<body>
    <form id="form1" runat="server">
    <div>
        <asp:ContentPlaceHolder ID="main" runat="server">
        </asp:ContentPlaceHolder>
    </div>
    </form>
</body>
</html>
```

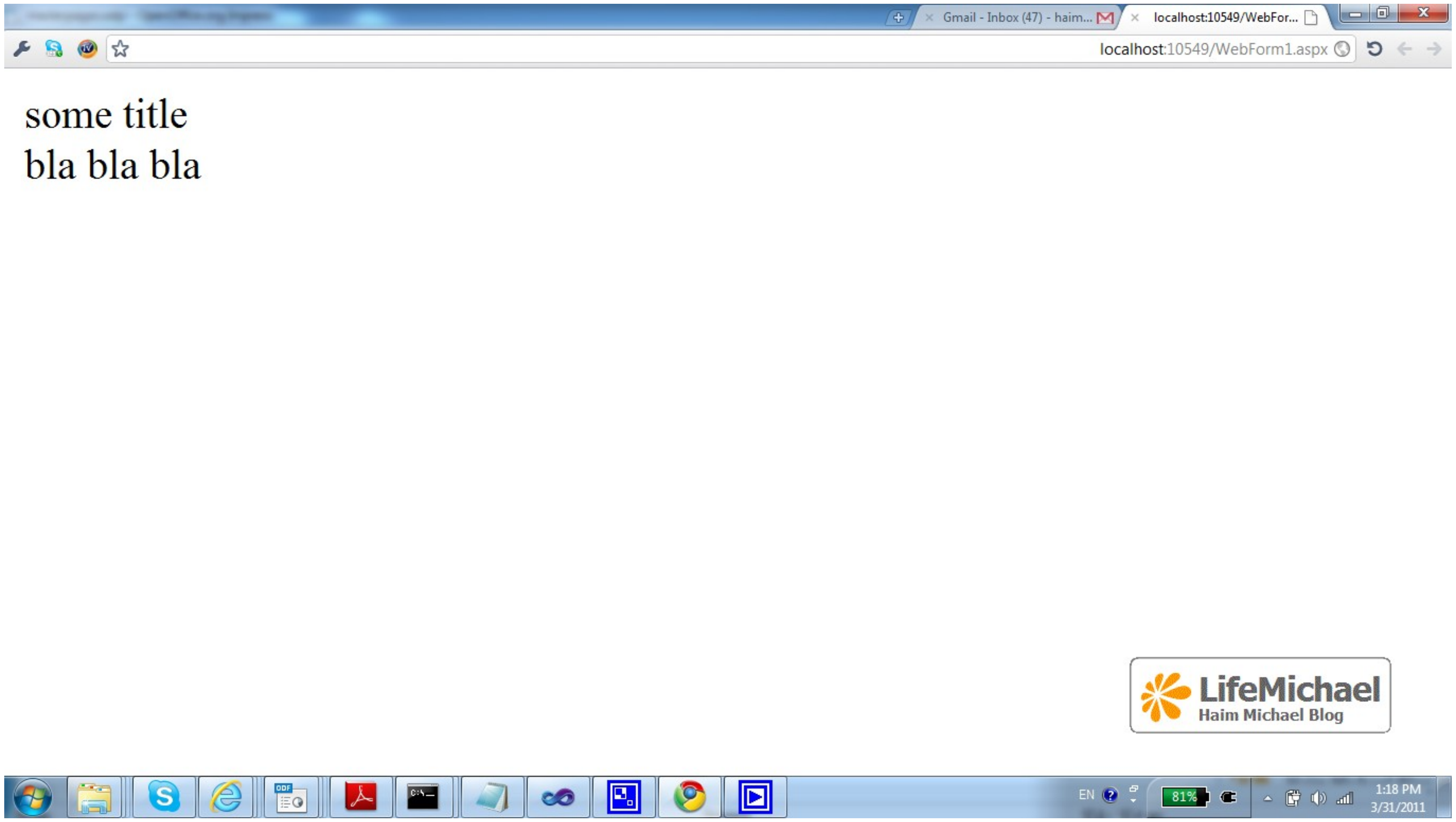
Nesting Master Pages

```
<%@ Master Language="C#" MasterPageFile="~/Site1.Master"
AutoEventWireup="true" CodeBehind="NestedMasterPage1.master.cs"
Inherits="WebApplication20.NestedMasterPage1" %>
<asp:Content ID="nestedhead" ContentPlaceHolderID="head" runat="server">
some title
</asp:Content>
<asp:Content ID="Content1" ContentPlaceHolderID="main" runat="server">
    <asp:ContentPlaceHolder ID="nestedmain" runat="server">
    </asp:ContentPlaceHolder>
</asp:Content>
```

Nesting Master Pages

```
<%@ Page Title="" Language="C#"
MasterPageFile="~/NestedMasterPage1.master" AutoEventWireup="true"
CodeBehind="WebForm1.aspx.cs" Inherits="WebApplication20.WebForm1" %>

<asp:Content ContentPlaceHolderID="nestedmain" runat="server">
bla bla bla
</asp:Content>
```

Nesting Master Pages



some title
bla bla bla