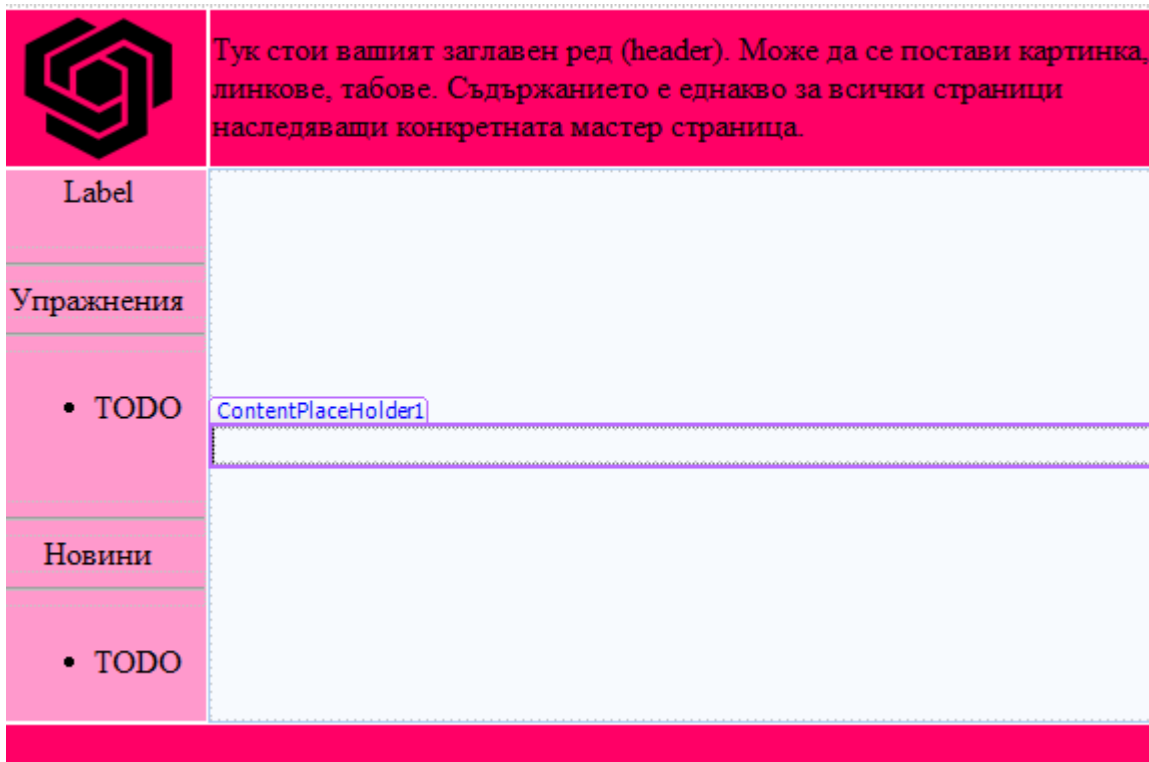


Упражнение 4 - Master Pages и навигация.

Задача

1. Създайте нова веб страница, чрез File -> New -> Web Site (C# language).
2. Изтрийте Default.aspx страница заедно с прилежащия и .cs файл.
3. Добавете Master страница към проекта си с Add New Item -> Master Page. Забележете, че разширението на създавания файл е .master.
4. Създайте семпъл дизайн на мастер страницата наподобяващ долната картинка. За позициониране на компонентите използвайте таблица (може и div).



5. Добавете aspx страница (Add New Item -> Web Form). В диалоговия прозорец за добавяне маркирайте „Select master page“ и след това изберете вече създадената от вас мастер страница.
6. Стартирайте с CTRL+F5 и вижте резултата. Визуализира се страница, съдържаща само компонентите на своята мастер страница, но без никакво друго допълнително съдържание.
7. Добавете някакъв текст или контроли към вашата Content страница. Тествайте промените.
8. Направете така, че при зареждане на страницата ви да в Label1 да се изписва текущото време. Използвайте метода `DateTime.Now.ToString("dddd, MMMM dd")` ;
9. Добавете още 2 страници (Page1.aspx и Page2.aspx), използващи готовата мастер страница. Сложете различаващ се текст в двете.

10. Добавете Menu контрола от раздела Navigation на Toolbox-а. Настройте я така, че да осигури връзка към страниците на вашия web site (трябва създадете елементи и да уточните стойностите на свойствата им Text и NavigateUrl).
11. Стартирайте приложението и наблюдавайте направените промени.

Съдържание:

- [Създаване на оформление с помощта на Master Pages](#)
 - [Master и Content страници](#)
 - [URL ребазирание в Master Page](#)
 - [Достъп до Master Page от кода](#)
 - [Вложени Master Pages](#)
- [Създаване на йерархична навигация на сайт](#)
 - [Създаване на карта на сайта \(Site Map\)](#)
 - [Употреба на Site Navigation контроли](#)
- [Задача](#)

Създаване на оформление с помощта на Master Pages

Master Page ви дава възможността да определите стила на вашата страница и кодът, описващ това, да се съдържа в един общ файл. Master Page е страница, която съдържа настройки и контроли, които следва да се разпределят в няколко страници от вашия сайт. Например, ако всичките ви страници, трябва да имат еднакви банери в хедъра и футъра, или едно и също навигационно меню, те могат да се дефинират в Master страницата веднъж, и след това всяка страница, свързана с тази Master страница ще наследи тези общи елементи. Предимството от определяне на хедър, футър и навигация в Mater Page е, че тези елементи трябва да бъдат дефинирани само веднъж, вместо да се повтаря код в страниците от вашия сайт.

Master и Content страници

Създаването на Master страницата е точно като създаване на нормалната страница. Master страниците могат да съдържат за markup, контроли, код или всяка комбинация от тези елементи. Въпреки това, Master страницата може да съдържа особен вид контрол, наречен ContentPlaceHolder контрол. ContentPlaceHolder определя полето от Master страница, което може да бъде заменено със съдържание от страница, свързана със съответната Master страница. ContentPlaceHolder може да съдържа и стандартно информация, в случаите, когато не е необходимо производната страница да променя това съдържание. Синтаксисът на ContentPlaceHolder контрола е даден по-долу:

```
<%-- ContentPlaceHolder control --%>
```

```
<asp:contentplaceholder id="FlowerText" runat="server"/>
```

```
<%-- ContentPlaceholder with default content --%>

<asp:contentplaceholder id="FlowerText" runat="server">

  <h3>Welcome to my florist website!</h3>

</asp:contentplaceholder>
```

За да се различава Master страницата от нормалната такава се запаметява с разширение .master на файла. Една страница може да използва дадена Master Page чрез дефиниране на MasterPageFile атрибут от своята Page директива както е показано по-долу. Страницата, която се асоциира Master Page се нарича Content Page.

Content страница може да дефинира Content контроли, които да предефинират contentplaceholder секциите в Master Page. Content контрола се асоциира с точно определен ContentPlaceholder контрол, чрез неговото ContentPlaceholderID свойство. Content страницата може да съдържа само markur и контроли вътре в Content контролите; сама по себе си тя не може да има съдържание от високо ниво. Въпреки това има директиви или server-side код.

```
<% @ Page MasterPageFile="Site.master" %>
```

```
<asp:content id="Content1" contentplaceholderid="FlowerText" runat="server">

  With sunshine, water, and careful tending, roses will bloom several times in a season.

</asp:content>

<asp:content id="Content2" contentplaceholderid="FlowerPicture" runat="server">

  <asp:Image id="image1" imageurl="~/images/rose.jpg" runat="server"/>

</asp:content>
```

Следващият пример демонстрира връзката между Master и Content страници. В този случай Master страницата дефинира две ContentPlaceholder региона, именувани FlowerPicture и FlowerText заедно с някаква стандартна информация в тях. Отделните content страници в този сайт наследяват оформлението от Master страницата, но предефинират стандартното съдържание за именуваните ContentPlaceholder региони с тяхно собствено съдържание. Вземете в предвид, че Default.aspx страницата в този сайт не дефинира Content контроли, а само наследява стандартното съдържание от Master страницата.

Пример: Master и Content страници



URL ребазирание в Master Page

Нещо, което може да забележите в предходния пример за това е, че има няколко места в Master страницата, които се отнасят до URL ресурси като изображения или стилове или препратки към страници относителен път, като например:

```
<head>

  <link rel="stylesheet" href="StyleSheet.css" type="text/css" />

</head>

...

<a href="daffodil.aspx">Daffodil</a>

...


```

Това работи добре, когато Master Page и Content Page са в една и съща директория, но когато Content Page е на физически различно местоположение, относителния път няма да бъде верен. За да се реши този проблем може да се използва някои от следните подходи:

- Използване на абсолютни URL пътища в Master Page, например ``
- Използване на относителни или свързани с приложението URL-та в сървърни контроли вместо статичен маркър, например `<asp:Image ImageUrl="~/images/banner.gif" runat="server" />`

Следващият пример демонстрира тази техника. Content страниците са преместени в поддиректория "Pages" под тази в която се съдържа Master страницата. Тя е променена в насока използване на сървърни контроли в замяна на HTML:

```
<head runat="server">

  <link rel="stylesheet" href="StyleSheet.css" type="text/css" />

</head>

...

<a id="A1" href="pages/daffodil.aspx" runat="server">Daffodil</a>
```

...

```
<asp:Image ID="Image1" AlternateText="Water Lillies"  
ImageUrl="~/Images/Waterlilies.jpg" runat="server"/>
```

Пример: URL свързване в Master Pages



Достъп до Master Page от кода

Възможно е Content страницата програмируемо да достъпва Master страница. Content страницата създава строго-типизирана препратка към Master страницата използвайки `<% @ MasterType%>` директива, като се посочва виртуален път към Master страница:

```
<%@ MasterType VirtualPath="Site.master" %>
```

Тогава Content страницата може да се обърне към Master страницата използвайки Master свойството на класа Page:

```
Master.FooterText = "This is a custom footer"  
Dim ad As AdRotator = Master.FindControl("MyAdRotator")
```

В горния пример `FooterText` е `public` свойство видимо в Master страницата, докато `MyAdRotator` е контрол в Master страницата. Следващия пример демонстрира този код в действие:

Пример: Достъп до Master Pages от кода



Вложени Master Pages

Content страниците от своя страна могат да бъдат също Master страници. Това означава, че е възможно една Master страница да произхожда от друга. Например, може да имате Master страница от най-горно ниво, която определя общия изглед на сайта - Header / Footer и навигация, а след това отделни Master страници, които произхождат от горната, за да се определи различен вид за различните подраздели на сайта. Content страници тогава ще произлизат от съответната master страница за раздел. Следващият пример показва тази идея, като разделя цветарския сайт в две секции, Annuals и Perrennials.

Пример: Вложени Master Pages



Създаване на йерархична навигация на сайт

Уеб сайтовете често трябва да показват данни за навигация, която да ръководи потребителите в сайта. Включените възможности за навигацията в ASP.NET позволяват на разработчиците да определят лесно навигационните данни и да изобразяват потребителски интерфейс въз основа на тази информация. Site Navigation API е абстракция за достъп до данни за навигация на сайта. API съхранява навигационните данни в XML файл и ги представя като съвкупност от SiteMapNode класове.

Навигационните сървърни контроли включват Menu, TreeView, SiteMapPath и SiteMapDataSource. Тези контроли са изградени на върха на Site Navigation класовете и работят със и изобразяват навигационни данни, без да се интересуват от специфични въпроси за съхранение на данни. Menu и Treeview контролите може също да използват данни от XML файлове и от контролът XMLDataSource.

Създаване на карта на сайта (Site Map)

Навигационните структури са представени във файлове Web.sitemap. Във файла карта, който можете да видите по-долу, е показана навигационна структура на един сайт. Файла Web.sitemap съдържа един елемент <siteMap> от най-високо ниво. Вложен в <siteMap> елемента е най-малко един <siteMapNode> елемент. Винаги трябва да има една главен <siteMapNode> в рамките на една карта на сайта. Site Navigation изисква единствен корен <siteMapNode> за да гарантира, че при изкачването по йерархията от възли ще се стигне до един добре познат възел. Под корена на един <siteMapNode> могат да се влагат колкото е необходимо <siteMapNode> елементи. Дълбочината на влагане на <siteMapNode> елементите също е неограничена.

Един <siteMapNode> елемент обикновено съдържа атрибутите Url, Title и Description. Атрибутът Url може да покаже виртуален път, който съответства на страница в приложението ви. Той може да съдържа и пътища към страници в други приложения, или URL адреси, сочещи напълно различни уеб сайтове. В примера по-долу всички Url атрибути сочат към страници от едно приложение. Атрибутът Title се използва като показва текст свързан с навигационните данни. Например, контролът SiteMapPath използва Title атрибута за да съставя текста на хипервръзките в контролата. Ако Description атрибутът е налице, сървърните контроли могат да използват тази информация, за да покажат подсказки или ALT текст. Разработчиците могат да добавят собствени атрибути към <siteMapNode> и тези атрибути ще бъдат на разположение чрез стандартния индексатор на класа SiteMapNode. За информация относно други атрибути поддържани от <siteMapNode> елемента, моля, вижте .NET Framework документацията.

Пример: Web.sitemap



Употреба на Site Navigation контроли

Един лесен начин за предоставяне на навигацията във вашата уеб страница е да се използва един от графичните контроли за навигация: SiteMapPath, TreeView and Menu.

- SiteMapPath - Основно контрол, който извлича текущата страница и показва йерархията. Това дава възможност на потребителите да навигират обратно към други страници в йерархията. SiteMapPath работи изключително с SiteMapProvider, който може да бъде определен чрез съответния контрол.
- TreeView - Дава възможност вертикалният потребителски интерфейс да разширява или свива избрани възли на дадена уеб страница, както и check box избор на елементи. TreeView контролата поддържа декларативни и програмни настройки на елементите с данни и контролите за източника на данни. Ако се използва SiteMapDataSource контрол, databinding се извършва автоматично.
- Menu - Осигурява хоризонтален или вертикален потребителски интерфейс, който изкарва допълнителни подменюта, когато потребител посочи определен елемент . Контролът Меню поддържа декларативни и програмни настройки на елементите с данни и контролите за източника на данни. Ако използвате SiteMapDataSource контрол, се извършва автоматично.

По-долу е представена таблица изтъкваща някои от разликите между TreeView и Menu за да ви помогне при избора на правилната контрола за конкретен сценарий.

Feature	Menu	TreeView
Expansion	PopOut	Expand in place
Download on Demand	No	Yes
Checkboxes	No	Yes
Templates	Yes	No
Layout	Horizontal & Vertical	Vertical
Style Options	Yes	Yes
Selection Model	Static, Dynamic Level	Level or Parent/Root/Leaf, or by data item

В следващия пример се наблюдават TreeView и Menu контроли, конфигурирани с различни стилни свойства за различни полета и действия (например NodeStyle и HoverNodeStyle). TreeView и Menu контролите се свързват към един и същ SiteMapDataSource контрол на страницата. SiteMapPath контролът достъпва същите данни чрез стандартния SiteMapProvider, който е настроен в Web.config файла.

Пример: Навигационни контроли

