



**TARUMT**  
TUNKU ABDUL RAHMAN  
UNIVERSITY OF  
MANAGEMENT AND TECHNOLOGY



**Premier  
Digital Tech  
Institution™**



# Server Controls and Site Design

## Chapter 2

# What Are You Going To Learn?

---

- At the end of this lesson, you will be able to:
  - explain ASP.NET Development Model
  - Identify and use the appropriate server controls to create the ASP.NET Web forms.
  - Use master pages and site map for a Website.

# ASP.NET Development Models:

---

- ASP.NET supports three different development models:
  - Web Pages
  - MVC (Model View Controller)
  - Web Forms (this is covered in our syllabus).

# ASP.NET Web Pages

---

- Simplest ASP.NET model
- Built around single web pages
- Similar to PHP and classic ASP
- Use server scripting (called Razor) with VB or C#
- Provided with built-in templates and helpers for database, video, graphics, social media, etc.

# PHP Example

```
<h1>Sample PHP Web Pages</h1>  
<?php  
    echo "Today is " . date("Y/m/d");  
?>
```

# ASP Example

```
<h1>Sample ASP.NET Web Pages</h1>  
<p>Today is <% response.write(date()) %></p>
```

# ASP.NET Web Pages Example

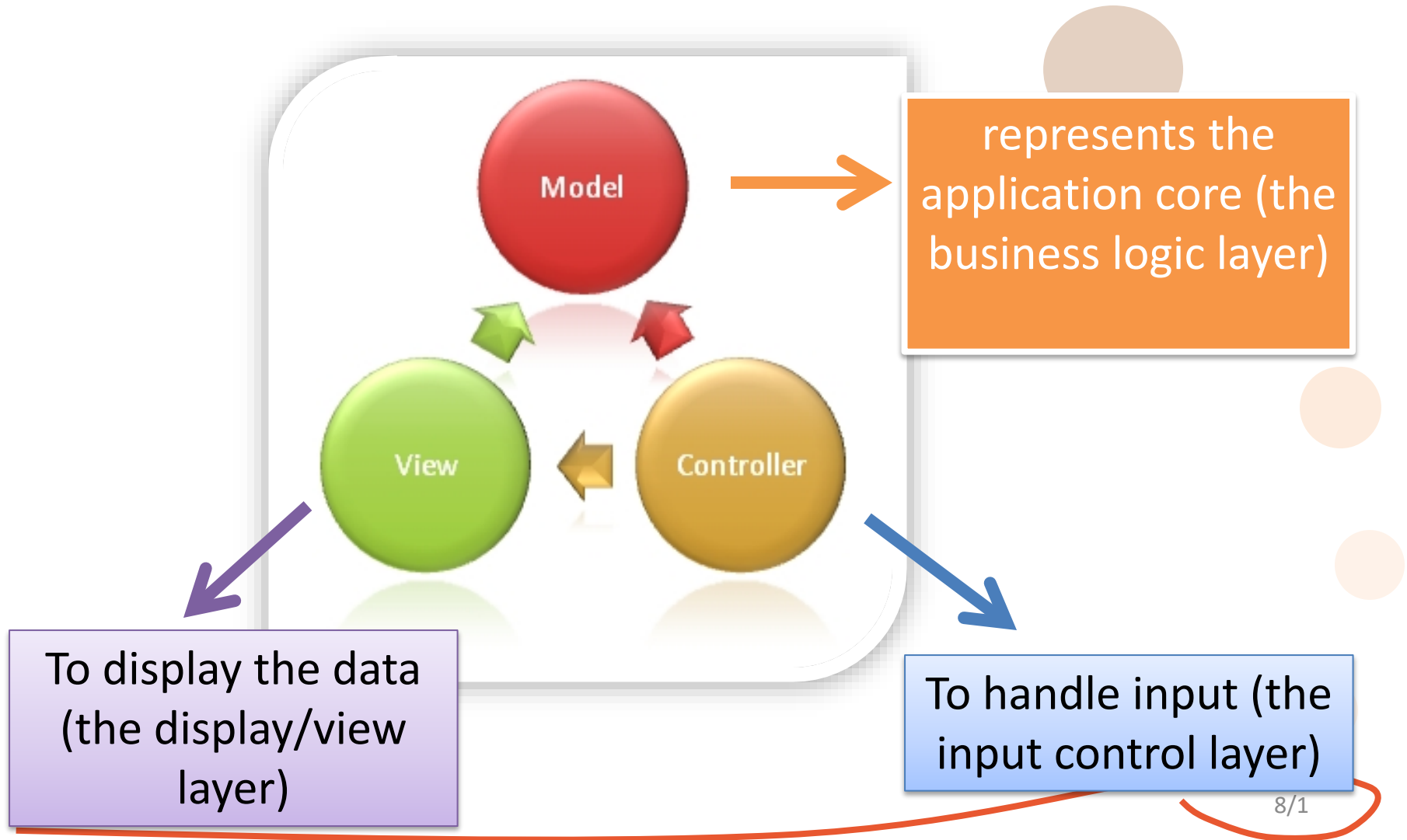
```
<html>
  <body>
    <h1>Sample ASP.NET Web Pages</h1>
    <p>Today is @DateTime.Now</p>
  </body>
</html>
```

Razor is a markup syntax that lets you embed server-based code (Visual Basic or C#) into web pages

# Web Pages with Database

```
@{ var db = Database.Open("SmallBakery");  
  var query = "SELECT * FROM Product"; }  
<html>  
<body>  
<h1>Small Bakery Products</h1>  
<table border="1" width="100%">  
<tr><th>Product</th> <th>Price</th> </tr>  
  @foreach(var row in db.Query(query))  
  {  
    <tr> <td>@row.Name</td><td>row.Price</td> </tr>  
  }  
</table>  
</body>  
</html>
```

# Model View Controller (MVC)



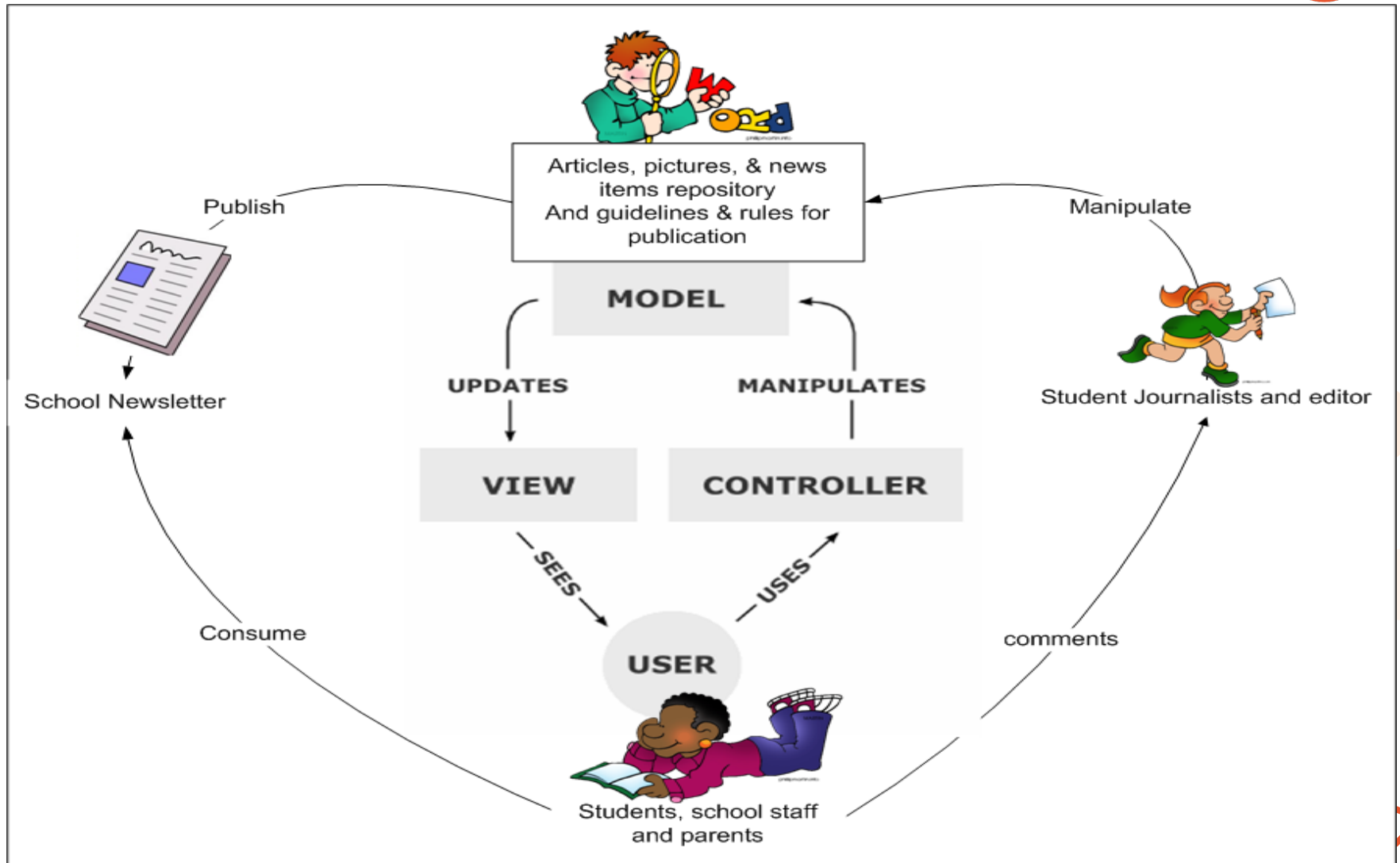


# ASP.NET MVC

---

- MVC is a framework for building web applications using a MVC (Model View Controller) design.
- Visual Web Developer is a development tool tailor made for ASP.NET MVC (and Web Forms).

# Model-view-controller - concept



# ASP.NET Web Forms

---

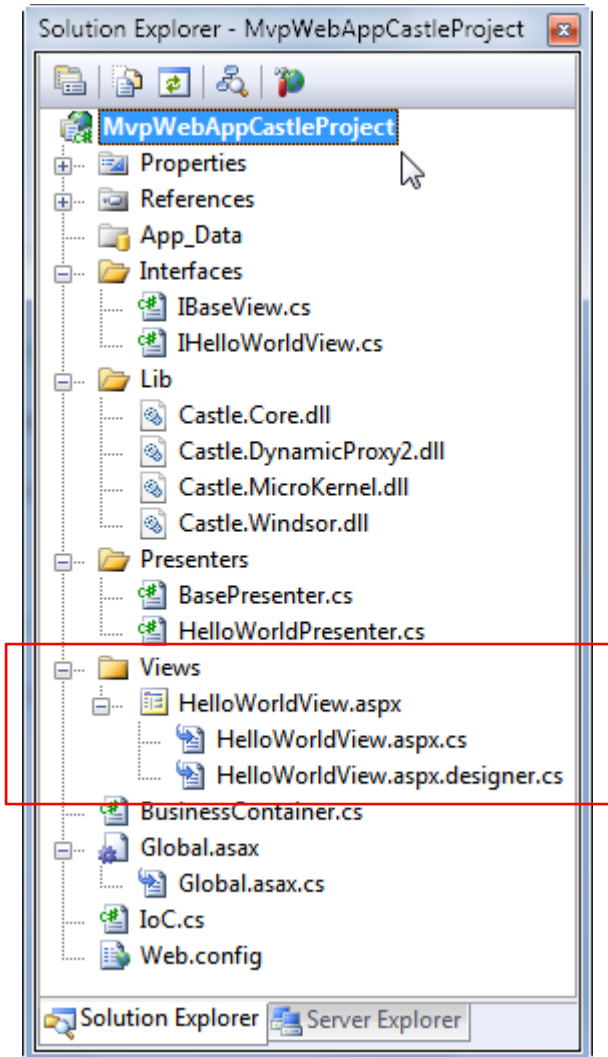
- The traditional ASP.NET and oldest event-driven development model
- Web pages with added server controls, server events, and server code.

# ASP.NET Web Forms

---

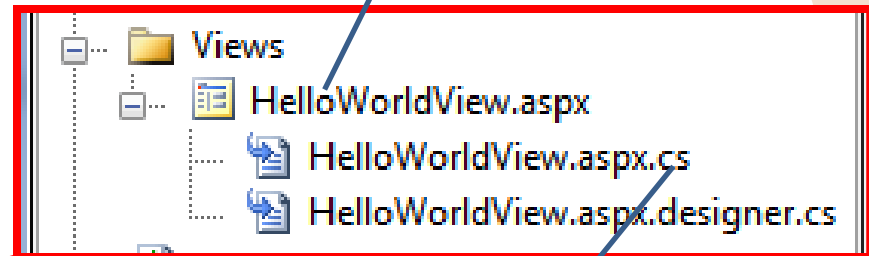
- refers to the grouping of two distinct blocks of code:
  - HTML template – presentation of the Web form on the browser.
  - ASP.NET code – holds a script containing the Web form's processing logic.

# Web Forms (Physical view)



Static HTML + Server controls

Presentation Layer



server code in C# (also known as **code-behind**)

To create dynamic content

# Web Form Syntax in the Presentation Layer

```
<body>
```

```
  <form ID="MyForm" runat="server">
```

... *here we place the server controls...*

```
  </form>
```

```
</body>
```

# Benefits of ASP.NET Server Controls

## Intuitive object model

- Expose the HTML elements of a page in an intuitive **object model**.

## Enable view state

- Automatically retain the value of their properties by participating in **view state**.

## Separate design from logic

- Enable you to cleanly **separate** the **design** content of a page **from** the application **logic**.

## Browser compatibility

- Enable you to maintain **browser compatibility** while still supporting advanced browser features.

# Server Controls and HTML Tags

ASP.NET Web Control	Similar HTML Form Tag
<code>&lt;asp:Label&gt;</code>	<code>&lt;Span&gt;</code> , <code>&lt;Div&gt;</code>
<code>&lt;asp:ListBox&gt;</code>	<code>&lt;Select&gt;</code>
<code>&lt;asp:DropDownList&gt;</code>	<code>&lt;Select&gt;</code>
<code>&lt;asp:TextBox&gt;</code>	<code>&lt;Input Type="Text"&gt;</code>
<code>&lt;asp:RadioButton&gt;</code> and <code>&lt;asp:RadioButtonList&gt;</code>	<code>&lt;Input Type="Radio"&gt;</code>
<code>&lt;asp:CheckBox&gt;</code> and <code>&lt;asp:CheckBoxList&gt;</code>	<code>&lt;Input Type="CheckBox"&gt;</code>
<code>&lt;asp:Button&gt;</code>	<code>&lt;Input Type="submit"&gt;</code>



# HTML control, HTML Server control and Server control

	HTML Control	Server Control
	<code>&lt;input type="text" /&gt;</code>	<code>&lt;asp:TextBox ID="txtName" runat="server" /&gt;</code>
<b>Processed by whom?</b>	By web browser	By server
<b>Why use it?</b>	Basic control rendering and client-side scripting	<ul style="list-style-type: none"><li>- Allows server to intervene and process.</li><li>- Enables view state which allows value to be retain even after page is refreshed.</li><li>- Circumvent browser compatibility issue.</li></ul>
<b>Limitations</b>	Need to consider browser compatibility	Must be processed by a server

# ASP.NET Server Controls Syntax

- All server controls must have two attributes
  - **runat** and **ID**

- E.g.:

```
<asp:Label id="lblMyLabel"  
runat="server">Sale Ends May  
2nd</asp:Label>
```

# Example with Code

1 **Name** \*:

Time Stamp : 1/1/0001 12:00:00 AM ex.: mm/dd/yyyy hh:mm PM

2 **Time Stamp** : 1/1/0001 12:00:00 AM ex.: An Overridden example

Enum : One ▾

3 **Number of Types** \* : ☒ One ☐ Two ☐ Three ☐ Four

Html :

Is Needed : ☐ **CheckBox or CheckBoxList?**

Integer Range Value : 0

Submit

```
<asp:RadioButtonList id="radType" runat="server">
```

```
<asp:TextBox id="txtHtml" rows="2" TextMode= "MultiLine" runat="server" />
```

```
<asp:ListItem value="Three" />  
<asp:ListItem value="Four" />
```

```
<asp:TextBox id="txtValue" runat="server" Text="0"/>
```

# Questions

---

- Suggest appropriate ASP.NET server controls that you should use in order to **obtain** the following data from a user:
  1. Name
  2. Gender
  3. Address
  4. Date of birth
  5. Favourite TV Channel



**TARUMT**  
TUNKU ABDUL RAHMAN  
UNIVERSITY OF  
MANAGEMENT AND TECHNOLOGY



**Premier  
Digital Tech  
Institution™**



# Site Design

## Master Page and Site Map

# Creating a Consistent Look and Feel Site

---

- A site benefits from a consistent look and feel, which generally includes the following:
  - A common header and menu system for the entire site.
  - A bar on the left side of the page offering some page navigation options.
  - A footer providing copyright information and a secondary menu for contacting the webmaster.

# Creating a Consistent Look and Feel Site

---

## Master Pages

defines the layout to be used by all pages based on the Master.

Essential elements (e.g. Header, menu) will be presented on every page

# ASP.NET Master Page



**Master file "A.master"**

```
<%@ Master %>
```

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

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



**Content file "A.aspx"**

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



```
<asp:Content runat=server  
ContentPlaceHolderId="Main" >  
Content here</asp:Content>
```



```
<asp:Content runat=server  
ContentPlaceHolderId="Footer" >  
Content here</asp:Content>
```

**Resulting  
Page**





# Advantages of Master Page

---

- Easy to create
  - You simply apply the same Master Page to the new content page.
- Easy to maintain
  - If you decide to completely modify the design of your website, you can modify just a single Master Page to change the appearance of all the pages in your application.

# Provide clear navigation mechanisms

---

- orientation information, navigation bars, a site map, etc. should be provided
  - to increase the likelihood that a person will find what they are looking for at a site.

# Site Map

---

- It is an overview of the pages within a website.
- It is a list of pages of a web site accessible to Web crawlers or users
- This allows visitors to quickly jump to any section of a website listed in the site map.
- This also gives visitors a good overall picture of how the site is organized and clearly defines all the resources the website has to

# Designing a Site Map

---

- site maps can be organized in a variety of ways. Typically organized in hierarchical fashion, most use an outline form, with pages arranged by topic.
- Can include pages for major categories and subcategories of the website

# ASP.NET Site Map

- ASP.NET holds this information in an XML file named `Web.sitemap`. This file is used as the source of data for menu and navigation controls.

- Navigation controls use site map:

- TreeView
- SiteMapPath (breadcrumbs)
- Menu



# ASP.NET Site Map - Code

```
<?xml version="1.0" encoding="ISO-8859-1" ?>  
<siteMap>
```

```
<siteMapNode title="Home" url="/aspnet/w3home.aspx">  
  <siteMapNode title="Services">  
    <siteMapNode title="Training" url="/aspnet/training.aspx"/>  
    <siteMapNode title="Support" url="/aspnet/support.aspx"  
description="Online support" />  
  </siteMapNode>  
</siteMapNode>  
</siteMap>
```

Main node (first level)

First Category: Services (2<sup>nd</sup> level)

Child nodes under "Services" (3<sup>rd</sup> level)

# ASP.NET Site Map - Code

- Explanation of the three main attributes of `<siteMapNode>`:
  - **title** : The title to be displayed
  - **Description**: A longer description that associates with a node. It is the tooltip that will appears when you hover the node.
  - **url** : A URL that points to a page or other resource. Include this if you want it to be a hyperlink

# ASP.NET Site Map - Code

---

- Therefore the output of the code:



Remark: Underline the text if it is a hyperlink



# Question

- Write the Code in the `web.sitemap` for the following structure:
  - Home [Default.aspx]
    - Products
      - Health Care [healthcare.aspx]
      - Sports [sports.aspx]
    - Services
      - Contact Us [contact.aspx]
      - Redemption [redemption.aspx]

This shows a tooltip  
“contact your nearest  
store”

This shows a tooltip  
“gift redemption”

# Question

- Identify 2 different errors of the code below:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

```
<siteMap>
```

```
<siteMapNode title="Product" url= "" >
```

```
    <siteMapNode title="Health Care" url="healthcare.aspx ">
```

```
    <siteMapNode title="Supplements" url="supplements.aspx" >
```

```
</siteMapNode>
```

```
<siteMapNode title="Services">
```

```
    <siteMapNode title="Training" url="training.aspx " >
```

```
    <siteMapNode title="Support" url="support.aspx" >
```

```
</siteMapNode>
```

```
</siteMap>
```



Site Map: Site Map structure

**DEMO**

# Next Week

---

- Event-driven Programming
  - Page Event
  - Server Control Event