# Web Applications

## Objective Domain Matrix

<table>
<thead>
<tr>
<th>Skills/Concepts</th>
<th>MTA Exam Objectives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Understanding Web Page Development</td>
<td>Understand Web page development (4.1)</td>
</tr>
<tr>
<td>Understanding ASP.NET Application Development</td>
<td>Understand Microsoft ASP.NET Web Application development (4.2)</td>
</tr>
<tr>
<td>Understanding IIS Web Hosting</td>
<td>Understand Web hosting (4.3)</td>
</tr>
<tr>
<td>Understanding Web Services Development</td>
<td>Understand Web Services (4.4)</td>
</tr>
</tbody>
</table>
HTML

- Hypertext Markup Language (HTML) is the language used by Web servers and browsers to describe a Web page.
- An HTML page has two distinct parts: a header and a body.
- HTML tags define the structure and content of a page. Each starting tag has a matching ending tag. For example, the ending tag for `<html>` is `</html>`.
- The header is enclosed within the `<head>` and `</head>` tags.
- The body is enclosed within the `<body>` and `</body>` tags.
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title></title>
  </head>
  <body>
    <h1>Mimas Cassini</h1>
    The img tag is used to display the picture of a <a href="http://en.wikipedia.org/wiki/Mimas_(moon)">
    Saturn's moon</a>: <br/>
    <img height="400px" width="400px" alt="Mimas Cassini" src="http://upload.wikimedia.org/wikipedia/commons/b/bc/Mimas_Cassini.jpg"/>
  </body>
</html>
Mimas Cassini

The img tag is used to display the picture of a Saturn's moon.
Cascading Style Sheets

- Cascading style sheets (CSS) is a language that describes information about displaying a Web page.

- CSS enable you to store a Web page’s style and formatting information separate from the HTML code.

- HTML specifies what will be displayed whereas CSS specifies how that material will be displayed.

- When used effectively, CSS is a great tool for increasing site-wide consistency and maintainability.
body
{
    font-family: Verdana;
    font-size: 9pt;
}
div
{
    color: Red;
}
.block
{
    background-color: Yellow;
    border-color: Blue;
    border-width: thin;
    border-style: outset;
    font-family: Arial;
}
Using the CSS File

<!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>
  <link href="styles.css" rel="stylesheet" type="text/css" />
  <title>Understanding CSS</title>
</head>

<body>
  Sample body text<br />
  <div>Sample DIV text</div>
  <div class="block">Sample DIV text with block class</div>
  <span class="block">Sample SPAN text with block class</span>
</body>
</html>
JavaScript

- JavaScript is a client-side scripting language.
- JavaScript runs inside Web browsers. JavaScript is supported by all popular Web Browsers.
- Ajax is shorthand for “Asynchronous JavaScript and XML.” Ajax uses JavaScript extensively in order to provide responsive Web applications.
- The ASP.NET AJAX framework lets you implement Ajax functionality on ASP.NET Web pages.
Understanding JavaScript

```html
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/1999/xhtml">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <title>Understanding JavaScript</title>
        <script type="text/javascript" language="javascript">
            var username = prompt("Please enter your name");
            var message = "Hello, " + username + ". Your name has ";
            var namelen = username.length;
            if (namelen > 5)
                message = message + "more than ";
            else if (namelen < 5)
                message = message + "less than ";
            else
                message = message + "exactly ";
            message = message + "5 characters."
            alert(message);
        </script>
    </head>
    <body>
    </body>
</html>
```
Client-Side vs. Server-Side Programming

- Client-side programming refers to programs that execute completely on a user’s local computer.
  - Examples: Windows Forms application, JavaScript code that executes within a Web browser.

- Server-side programming refers to programs that are executed completely on a server and make use of the server’s computational resources.
  - Examples: Web applications and Web services.

- Hybrid applications use both client- and server-side programming. Ajax applications use a mix of server-side programming and client-side code to create interactive and highly responsive Web applications.
ASP.NET

• ASP.NET is the part of the .NET Framework that enables you to develop Web applications and Web services.

• The ASP.NET infrastructure has two main parts:
  – A set of classes and interfaces that enables communication between the Web browser and Web server. These classes are organized in the System.Web namespace.
  – A runtime process, also known as the ASP.NET worker process (aspnet_wp.exe), that handles the Web request for ASP.NET resources.
ASP.NET Page Execution

- The ASP.NET worker process (aspnet_wp.exe) fulfills the request for ASP.NET page execution.
- The ASP.NET worker process compiles the .aspx file into an assembly and instructs the Common Language Runtime (CLR) to execute the assembly.
- When the assembly executes, it takes the services of various classes in the .NET Framework class library to accomplish its work and generate response messages for the requesting client.
- The ASP.NET worker process collects the responses generated by the execution of the Web page and creates a response packet.
# ASP.NET Page Life Cycle

<table>
<thead>
<tr>
<th>Event</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PreInit</td>
<td>Several page properties, such as Request, Response, IsPostBack, and UICulture, are set at this stage.</td>
</tr>
<tr>
<td>Init</td>
<td>During the initialization stage, all the controls on the page are initialized and made available.</td>
</tr>
<tr>
<td>Load</td>
<td>If the request is a postback, the load stage is used to restore control properties with information from view state and control state.</td>
</tr>
<tr>
<td>PreRender</td>
<td>This stage signals that the page is just about to render its contents.</td>
</tr>
<tr>
<td>Unload</td>
<td>During the unload stage, the response is sent to the client and page cleanup is performed.</td>
</tr>
</tbody>
</table>
State Management

• The state of a Web page is made up of the values of the various variables and controls.
• State management is the process of preserving the state of a Web page across multiple trips between browser and server.
• State Management Techniques:
  – Client-side state management
  – Server-side state management
Client-Side State Management

- Client-side techniques use HTML code and the capabilities of the Web browser to store state information on the client computer.

- Common client-side state management techniques:
  - Query Strings
  - Cookies
  - Hidden Fields
  - View State
Query Strings

- Query strings stores the data values in the query string portion of a page URL. For example, the following URL embeds a key ("q") and its value ("television") in query string: [http://www.bing.com/search?q=television](http://www.bing.com/search?q=television).

- To retrieve the value of the key in an ASP.NET page, use the expression: Request.QueryString["q"].

- **QueryString** is a property of the Request object, and it gets the collection of all the query-string variables and their values.
Cookies are small packets of information that are stored by a Web browser locally on the user’s computer. Cookies are commonly used for storing user preferences.

To add a Cookie:
HttpCookie cookie = new HttpCookie("Name", "Bob");
cookie.Expires = DateTime.Now.AddMinutes(10);
Response.Cookies.Add(cookie);

To read a cookie:
if (Request.Cookies["Name"] != null)
{
    name = Request.Cookies["Name"].Value;
}
**Hidden Fields**

- Hidden fields contain information that is not displayed on a Web page but is still part of the page’s HTML code.

- Hidden fields can be created by using the following HTML element:
  
  `<input type="hidden">`

- The ASP.NET HTML Server control `HtmlInputHidden` also maps to this HTML element.
View State

- ASP.NET uses View State to maintain the state of controls across page postbacks.
- When ASP.NET executes a page, it collects the values of all nonpostback controls that are modified in the code and formats them into a single encoded string. This string is stored in a hidden field in a control named __VIEWSTATE.
- View State may increase the size of your page.
- View State is enabled by default by you can disable it either at the control level or at the page level.
Server-Side State Management

- Server-side state management uses server resources to store state information.
- Storing and processing session information on a server increases the server’s load and requires additional server resources to serve the Web pages.
- ASP.NET supports server-side state management at two levels:
  - Session State
  - Application State
Session State

- An ASP.NET application creates a unique session for each user who sends a request to the application.
- Session state can be used for temporarily store user data such as shopping cart contents.

- Reading from session:
  ```csharp
  if (Session["Name"] != null)
  {
      /* additional code here */
  }
  ```
- Writing to session:
  ```csharp
  Session.Add("Name", TextBox1.Text);
  ```
Application State

- Application state is used to store data that is used throughout an application.
- Application State is not user-specific.
- Application state can be accessed through the Application property of the Page class.
- The Application property provides access to the HttpApplicationState object.
- The HttpApplicationState object stores the application state as a collection of key-value pairs.
Internet Information Services

- Internet Information Services (IIS) is a Web server for hosting Web applications on the Windows operating system.
Web Site & Virtual Directories

- An IIS server uses the concepts of sites, applications, and virtual directories.

- A Web site (for example: www.northwind.com) is a container of applications and virtual directories.

- A virtual directory is an alias that maps to a physical directory on the Web server. For example, in the address www.northwind.com/orders, “orders” is a virtual directory.

- A virtual directory maps to a physical directory (for example, c:\inetpub\wwwroot\northwind\orders).
Creating a Virtual Directory

Add Virtual Directory

- Site name: Default Web Site
- Path: /

- Alias: orders
- Example: images
- Physical path: c:\project\northwind\orders
- Pass-through authentication

[Buttons: Connect as..., Test Settings..., OK, Cancel]
Deploying Web Applications

• Xcopy or FTP
For simple websites that require simply copying the files.

• Windows Installer
For complex websites that require custom actions during the deployment process. Windows Installer can create virtual directories, restart services, register components, etc.
Web Services

- Web services provide a way to interact with programming objects located on remote computers.
- Web services are based on standard technologies and are interoperable.

Key technologies:
- Hypertext Transmission Protocol (HTTP)
- Extensible Markup Language (XML)
- Simple Object Access Protocol (SOAP)
- Web Services Description Language (WSDL)
SOAP

- SOAP is the protocol that defines how remote computers exchange messages as part of a Web service communication.
- Message format: XML
  - XML is easier for otherwise non compatible systems to understand.
- Message transmission: HTTP
  - HTTP, they can normally reach any machine on the Internet without being blocked by firewalls.
WSDL

- WSDL stands for Web Services Description Language.
- A WSDL file acts as the public interface of a Web service and includes the following information:
  - The data types it can process
  - The methods it exposes
  - The URLs through which those methods can be accessed
Creating Web Services

- Inherit from System.Web.Services.WebService
- Mark the class with a WebService attribute
- Mark the methods with WebMethod attribute

```csharp
[WebService(Namespace = "http://northwindtraders.com")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
{
    [WebMethod]
    public string ToUpper(string inputString)
    {
        return inputString.ToUpper();
    }
}
```
Consuming Web Services

- Add a reference to the Web service by using the Add Web Reference dialog Box.
Consuming Web Services

- The proxy object allows you to invoke Web service methods.

protected void Button1_Click(object sender, EventArgs e)
{
    var webService = new textWebService.TextWebService();
    toLowerLabel.Text = webService.ToLower(TextBox1.Text);
    toUpperLabel.Text = webService.ToUpper(TextBox1.Text);
}
Recap

- Web Page Development
  - HTML, CSS, JavaScript
- Client-side vs. server-side programming
- ASP.NET Page Life Cycle
  - PreInit, Init, Load, PreRender, and Unload
- State Management
  - Query strings, cookies, hidden fields, viewstate
  - Session state, application state
- IIS Web Hosting
  - Web Sites, Virtual directories
- Web Services
  - SOAP, WSDL
  - WebService attribute, WebMethod attribute