RAMRAO ADIK INSTITUTE OF TECHNOLOGY, NERUL

LAB MANUAL

WEB TECHNOLOG Y

DEPARTMENT OF COMPUTER ENGG.

LIST OF EXPERIMENTS

Name of Faculty : **Ashwini Gudewar**	Term:**Jan 2009-May 2009**
Subject: **Web Technology**	Lab Session**:3 hrs/week**
Year & Semester : **T.E. Sem VI**	No. of Lectures per week: **3**
Branch : **Computers**	No of Weeks: **15**

Sr No.	**Title**
1.	To study and implement all basic HTML tags.
2.	To design HTML form
3.	To design HTML frames
4.	To implement cascaded style sheet
5.	Design bank transaction form using Javascript.
6.	Design calendar using Javascript.
7.	To implement XML document using DTD and XML schema
8.	Design student registration form using servlet.
9.	Implement database connection using ASP.
10.	Implement session management using ASP.

EXPERIMENT NO 1

Aim: To study and implement all basic HTML tags.

Theory:

HTML is a language for describing web pages.

- HTML stands for Hyper Text Markup Language
- HTML is not a programming language, it is a markup language
- A markup language is a set of markup tags
- HTML uses markup tags to describe web pages

HTML Elements : HTML documents are text files made up of HTML elements.HTML elements are defined using HTML tags.

HTML Tags :

- HTML tags are used to mark-up HTML elements.
- HTML tags normally comes in pairs like <b>and </b>.
- The first tag in pair is the start tag; the second tag is the end tag.
- The text between the start and end tags is the element content.
- Some HTML elements have a missing end tag.
- HTML tags are not case sensitive; <b> means the same as <B>.

Tag Attributes: Attributes can provide additional information about the HTML elements. Consider <body> tag with an added bgcolor attribute, we can set the background color of web page. For example, <body bgcolor=òredò> this will set the background color to red.

- Attributes always come in name/value pairs like name=òvalueò.
- Attributes are always specified in the start tag.

HTML Headings:

- Headings are defined with the <h1> to <h6> tags.
- <h1> defines the largest heading. <h6> defines the smallest heading.

HTML Paragraphs:

- Paragraphs are defined with the <p> tag.
- HTML automatically adds an extra blank line before and after a paragraph.

HTML Formatting Tags:

- HTML uses tags like <b> and <i> for formatting text in bold and italic form.
- The <br> tag is used when we want to end a line, but donôt want to start a new paragraph.
- The <br> tag is an empty tag. It has no closing tag.

Anchor Tag and Href attribute:

- HTML uses the <a> (anchor) tag to create a link to another document.
- An anchor can point to any resource on web for example an HTML page, an image, a sound file etc.
- The syntax of creating an anchor:
  <a href=òurlò> Text to be displayed </a>
- **HREF** stands for **H**ypertext **REF**erence. The href attribute is used to create a hypertext link.

The target Attribute:

- The **target attribute** defines **where** the linked document will be opened.

The name Attribute:

- The **name attribute** is used to create a named anchor.

The Image Tag and the Src Attribute :

- In HTML, images are defined with the <img> tag.
- The <img> tag is empty, which means that it contains attributes only and it has no closing tag.
- To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display on your page.URL points to the location where the image is stored.
- The syntax of defining an image:
  <img src="url">

The Alt Attribute :

- The alt attribute is used to define an "alternate text" for an image. The value of the alt attribute is an author-defined text:
  <img src="boat.gif" alt="Big Boat">
- The "alt" attribute tells the reader what he or she is missing on a page if the browser can't load images. The browser will then display the alternate text instead of the image. It is a good practice to include the "alt" attribute for each image on a page, to improve the display and usefulness of your document for people who have text-only browsers.

List Tags:

Tag	**Description**
<ol>	Defines an ordered list
<ul>	Defines an unordered list
<li>	Defines a list item
<dl>	Defines a definition list
<dt>	Defines a definition term
<dd>	Defines a definition description

Table Tags:

Tag	**Description**
<table>	Defines a table
<th>	Defines a table header
<tr>	Defines a table row
<td>	Defines a table cell
<caption>	Defines a table caption
<colgroup>	Defines groups of table columns
<col>	Defines the attribute values for one or more columns in a table
<thead>	Defines a table head
<tbody>	Defines a table body
<tfoot>	Defines a table footer

Conclusion:

Thus we have created a web page using different HTML tags.

EXPERIMENT NO 2

Aim: To design HTML form

Theory:

Forms are used to collect information from user and transmit that information to server for processing. It is an area that can contain form elements. Form elements are elements that allow the user to enter information like text fields, text area fields, drop-down menus, radio buttons, checkboxes, etc. in a form.

A form is defined with the <form> tag. For example:

```
<form>
	<input>
	</input>
</form>
```

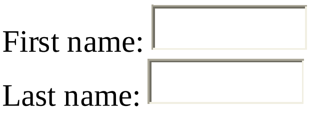
Input: The most used form tag is the <input> tag. The type of input is specified with the type attribute. The most commonly used input types are explained below.

Text Fields: Text fields are used when you want the user to type letters, numbers, etc. in a form. The width of the text field is 20 characters by default.

Consider the following example:

```
<form>
First name:
<input type="text" name="first name">
<br>
Last name:
<input type="text" name="last name">
</form>
```

It looks in a browser as shown below:



First name:
Last name:

Radio Buttons: Radio Buttons are used when user want to select one of a limited number of choices. Only one option can be chosen at a time.

Consider the following example:

```
<form>
```

```
<input type="radio" name="sex" value="male"> Male
<br>
<input type="radio" name="sex" value="female"> Female
</form>
```

It looks in a browser as shown below:

○ Male

○ Female

Checkboxes: Checkboxes are used when you want the user to select one or more options of a limited number of choices.

Consider the following example:

```
<form>
I have a bike:
<input type="checkbox" name="vehicle" value="Bike">
<br>
I have a car:
<input type="checkbox" name="vehicle" value="Car">
<br>
I have an airplane:
<input type="checkbox" name="vehicle" value="Airplane">
</form>
```

It looks in a browser as shown below:

I have a bike: □

I have a car: □

I have an airplane: □

The Form's Action Attribute and the Submit Button:

When the user clicks on the "Submit" button, the content of the form is sent to the server. The form's action attribute defines the name of the file where the user wants to send the contents of form. The file defined in the action attribute usually does something with the received input.

Consider the following example:

```
<form name="input" action="html_form_submit.asp" method="get">
 Username:    <input type="text" name="user">
<input type="submit" value="Submit">
</form>
```

Conclusion:

We have studied and implemented HTML form tags to create different forms required for our project.

EXPERIMENT NO 3

Aim: To design HTML frames

Theory:

With frames, you can display more than one HTML document in the same browser window. Each HTML document is called a frame, and each frame is independent of the others. Frames allow content and navigation to be separated from each other.It is used typically to present a left hand navigation menu and a top banner giving access to the main parts of the site.

A frame can contain interactive tables of the contents with links that when clicked display results in adjoining frame. Frames designed side by side permit queries to be passed and answered on the same page, with one frame holding the query frame and the other presenting the results.

Disadvantages :

- Frames often reduce the amount of usable space on the page.
- For low resolution monitors ,frames may be inconvenient
- It is difficult to print the content of all frames when compared to a normal web page.
- One of the main disadvantages of frames is that search engines will not deal with them properly.

The Frameset Tag:

- The <frameset> tag defines how to divide the window into frames
- Each frameset defines a set of rows or columns
- The values of the rows/columns indicate the amount of screen area each row/column will occupy

The Frame Tag:

- The <frame> tag defines what HTML document to put into each frame In the example below we have a frameset with two columns. The first column is set to 25% of the width of the browser window. The second column is set to 75% of the width of the browser window. The HTML document "frame_a.htm" is put into the first column, and the HTML document "frame_b.htm" is put into the second column:

```
<frameset cols="25%,75%">
  <frame src="frame_a.htm">
  <frame src="frame_b.htm">
</frameset>
```

The frameset column size value can also be set in pixels (cols="200,500"), and one of the columns can be set to use the remaining space (cols="25 %,*").

Attributes of Frame Tag :

1. src = òurlò :-
   Specifies the address of the document to be displayed in the frame.
2. name = òwindow nameò :-
   This is used to assign the name to frame so that it can be targeted by links in other documents.
3. frameborder = ò1/0ò :-
   When set to 1 a separator is drawn on every side next to another frame. Default value is 1. When set to 0 there is no separator.
4. marginwidth=òvalue in pixelò :-
   This is used when some margin for the frame is required horizontally.

5. marginheight=òvalue in pixelò :-
   This is used when some margin for the frame is required vertically.
6. noresize :-
   It means frame is not resizable by the user. It does not have any value.
7. scrolling = òyes/noò :-
   This indicates whether the frame should have scrollbars or not. Default value is yes.

Conclusion:

Frames are useful for displaying more web pages on same browser but the web developer must keep track of more HTML documents.

EXPERIMENT NO 4

Aim: To implement cascaded style sheet

Theory:

CSS (CASCADED STYLE SHEET) :

- **CSS** is a simple mechanism for adding style to web documents.
- Style sheet enables you to change the appearance and layout of the web page.
- Style sheet define how to display HTML elements.
- With plain Html we define the colors and sizes of the text and tables throughout our pages. If we want to change a certain element we will therefore have to work through the document and change it. Whereas with css we define the colors and sizes in styles. If we change a certain style it will change the look of entire site.
- Even multiple external style sheets can be referenced inside a single HTML document.

Syntax:

The CSS syntax is made up of three parts : a selector, a property and a value

```
selector {property: value}
```

The selector is normally the HTML element/tag you wish to define, the property is the attribute you wish to change, and each property can take a value. The property and value are separated by a colon, and surrounded by curly braces:

```
body {color: black}
```

If the value is multiple words, put quotes around the value:

```
p {font-family: "sans serif"}
```

If we wish to specify more than one property, we must separate each property with a semicolon. The example below shows how to define a center aligned paragraph, with a red text color:

```
p {text-align:center;color:red}
```

To make the style definitions more readable, you can describe one property on each line, like this:

```
p{
text-align: center;
color: black;
font-family: arial
}
```

Types Of CSS :

1.External Style Sheet :

- External CSS is a file tat contains only CSS code and is saved with a .css file extension.
- With an external style sheet we can change the look of an entire web site by changing one file.
- Each page must link to the style sheet using the <link> tag.
- The <link> tag goes inside the head section.
- We can use one external style sheet with multiple HTML files.

2.Internal Style Sheet :

- Internal style sheet is defined inside the head section by using the <style> tag.
- An internal style sheet should be used when a single document has a unique style.

3.Inline Style Sheet :

- Inline style sheet is also called as embedded style sheet.
- This is defined along with the HTML tags.
- Inline Style Sheet has more priority than Internal and External Style Sheet.

Conclusion:

CSS is used for specific web layout which is not possible with normal HTML tags. For example, h1-h6 tag with color, placing an image and paragraph at one line. We can use different style sheet to design web page as per the requirement.

EXPERIMENT NO 5

Aim: Design bank transaction form using JavaScript

Theory:

JavaScript is a scripting language. JavaScript is designed to add interactivity to HTML pages. Without a scripting language like JavaScript, web pages have no processing capabilities or any way to interact with the site visitor. For Example, although visitors can enter an information into forms, HTML only lets the browser send the data to the web server. HTML has no way of validating or doing anything else with the information the visitor types or selection the visitor makes on a form.

JavaScript can be used to validate form data before it is submitted to a server, this will save the server from extra processing. JavaScript is case-sensitive language. It treats everything as an object. Each object of JavaScript is having its own properties, methods and events.

Script is nothing but the set of instruction that web browser should follow. We can define script inside head section or inside body section or inside both the section. We can define multiple scripts in one HTML file.

From Event Handlers :

1. onSubmit Event Handler : It is executed when form is submitted using a submt <input> tag. The onSubmit event is used to validate ALL form fields before it is send to a server.

Below is an example of how to use the onSubmit event. The checkForm() function will be called when the user clicks the submit button in the form. If the field values are not accepted, the submit should be cancelled. The function checkForm() returns either true or false. If it returns true the form will be submitted, otherwise the submit will be cancelled:

```
<form method="post" action="xxx.htm"
onsubmit="return checkForm()">
```

2. onReset Event Handler : It is executed when a reset button is selected on a form. It is used to confirm that the user really wants to reset the contents of a form.

Conclusion:

Form validation is done using JavaScript.

EXPERIMENT NO 6

Aim: Design calendar using JavaScript.

Theory:

The Date Object :

The Date object is used to work with date and time. There are different ways to instantiating a Date object.

1. new Date() For e.g. var d=new Date()
2. new Date(milliseconds) For e.g. var d=new Date(500)
3. new Date(yy, mm, dd) for e.g. var d=new Date(09,3,18)
4. new Date(year, minute, date, hh, mm, ss)
   For e.g. var d=new Date(2009,03,15,10,30,00)

Different methods of Date object are listed below:

Methods	**Description**
getDate()	Returns the day of the month(from 1-31)
getDay()	Returns the day of the week (0-Sunday,1-Monday and so on)
getMonth()	Returns the month (0-Jan,1-Feb and so on)
getFullYear()	Returns year in full 4 digit format.
getYear()	Returns the year.
getHours()	Returns the hour of a day.
getMinutes()	Returns the minutes.
getSeconds()	Returns the seconds
getMilliseconds()	Returns the milliseconds
setFullYear(year)	Sets the year of the Date object
setYear(year)	Sets the year of the Date object. Year can be two digits.
setMonth(mm)	Sets the month of the Date object.
setDate(day)	Sets the day of the month.
setHours(hh)	Sets the hour of the Date object.
setMinutes(mm)	Sets the minutes of the Date object.
setSeconds(ss)	Sets the seconds of the Date object.
setMilliseconds(millis)	Sets the millisecond field of the Date object.

We can easily manipulate the date by using the methods available for the Date object. In the example below we set a Date object to a specific date (14th January 2010):

```
vary myDate=new Date();
myDate.setFullYear(2010,0,14);
```

And in the following example we set a Date object to be 5 days into the future:

```
var myDate=new Date();
myDate.setDate(myDate.getDate()+5);
```

Conclusion:

Calendar is designed using Date object of JavaScript.

EXPERIMENT NO 7

Aim: To implement XML document using DTD and XML schema.

Theory:

XML:

- XML stands for EXtensible Markup Language
- XML is a markup language much like HTML
- XML was designed to carry data, not to display data
- XML tags are not predefined. You must define your own tags
- XML is designed to be self-descriptive
- XML is a W3C Recommendation
- All XML Elements Must Have a Closing Tag

In HTML, you will often see elements that don't have a closing tag:

```
<p>This is a paragraph
<p>This is another paragraph
```

In XML, it is illegal to omit the closing tag. All elements **must** have a closing tag:

```
<p>This is a paragraph</p>
<p>This is another paragraph</p>
```

XML Tags are Case Sensitive:

XML elements are defined using XML tags.

XML tags are case sensitive. With XML, the tag <Letter> is different from the tag <letter>.

Opening and closing tags must be written with the same case:

```
<Message>This is incorrect</message>
<message>This is correct</message>
```

XML Elements must be properly nested:

In HTML, you will often see improperly nested elements:

```
<b><i>This text is bold and italic</b></i>
```

In XML, all elements **must** be properly nested within each other:

```
<b><i>This text is bold and italic</i></b>
```

In the example above, "Properly nested" simply means that since the <i> element is opened inside the <b> element, it must be closed inside the <b> element.

XML Documents Must Have a Root Element:

XML documents must contain one element that is the **parent** of all other elements. This element is called the **root** element.

```
<root>
  <child>
    <subchild>.....</subchild>
  </child>
</root>
```

XML Attribute Values must be quoted:

XML elements can have attributes in name/value pairs just like in HTML.

In XML the attribute value must always be quoted. Study the two XML documents below. The first one is incorrect, the second is correct:

```
<note date=12/11/2007>
<to>Tove</to>
<from>Jani</from>
</note>
```

```
<note date="12/11/2007">
<to>Tove</to>
<from>Jani</from>
</note>
```

The error in the first document is that the date attribute in the note element is not quoted.

Entity References:

Some characters have a special meaning in XML.
If you place a character like "<" inside an XML element, it will generate an error because the parser interprets it as the start of a new element.
This will generate an XML error:

```
<message>if salary < 1000 then</message>
```

To avoid this error, replace the "<" character with an **entity reference**:

```
<message>if salary &lt; 1000 then</message>
```

There are 5 predefined entity references in XML:

&lt;	<	less than
&gt;	>	greater than
&amp;	&	ampersand
&apos;	'	apostrophe
&quot;	"	quotation mark

Comments in XML:

The syntax for writing comments in XML is similar to that of HTML.
<!-- This is a comment -->

With XML, White Space is preserved:

HTML reduces multiple white space characters to a single white space:

HTML:	Hello my name is Tove
Output:	Hello my name is Tove.

With XML, the white space in your document is not truncated.

Conclusion:

There are many differences in XML when compared to HTML, for example in XML we can create user defined tags, opening and closing tags are must in all XML while it can be omitted in some tags of HTML.

EXPERIMENT NO 8

Aim: To design student registration form and database connection using servlet.

Theory:

Servlets are modules that extend request/response oriented servers. Servlets are protocol and platform independent server side components written in java. Most CGI scripting languages do not check for memory overflow and array bounds. Since servlets are written in Java, they do not suffer from such drawbacks. They are used to handle multiple request from clients simultaneously. They act as a middle layer between a request coming from a web browser and applications running on an Application server.
The javax.servlet package provides interface and classes for writing servlets

Functionality of a Servlet s:

- Manage and store data submitted by an HTML form.
- Provide dynamic response to HTTP request for e.g. returning a result of a database query to a client.
- Read data entered by a client and looks for a additional information that is embedded in the HTTP request. This information could relate to browser capabilities , cookies, host name of the requesting client.
- Prepare a response and send it back to the client by setting appropriate HTTP response parameters such as type of document, cookies etc. The response could be in text form , binary form or a zip file.
- Servlets can handle multiple requests concurrently and can synchronize requests. This allows servlets to support systems such as on-line conferencing.

The Servlet Interface:

All servlets implement this interface either directly or by extending a class that implements it .The Servlet interface declares, but does not implement methods that manage the servlet and its communication with clients.

Client Interaction:

When a servlet accepts a call from a client, it receives two objects:

- A ServletRequest, which encapsulates the communication from the client to the server.
- A ServletResponse, which encapsulates the communication from the Servlet back to the client.

ServletRequest and ServletResponse are interfaces defined by the javax,servlet package.

The ServletRequest Interface:

The ServletRequest interface allows the servlet access to:

- Information such as the names of the parameters passed in by the client, the protocols being used by the client and the names of the remote host that made the request and the server that received it.

- Servlets use the input stream to get data from clients that use application protocols such as the HTTP, POST and PUT methods.

Interfaces that extend ServletRequest interface allow the servlet to retrieve more protocols-specific data.For example, the HttpServletRequest interface contains for accessing HTTP specific header information.

The ServletResponse Interface:

The ServletResponse interface gives the servlets methods for replying to client.

- Allows the servlet to set the content length and MIME type of the reply.
- Provides an output stream, ServletOutputStream and a writer through which the servlet can send the reply data.

Interfaces that extend the ServletResponse interface give the servlet more protocol specific capabilities. For e.g. the HttpServletResponse interface contains methods that allow the servlet to manipulate HTTP specific header information.

Conclusion:

The advantage for servlet with regard to database connectivity is that servlet life cycle allows servlet to maintain open database connections.

EXPERIMENT NO 9

Aim: Implement database connection using ASP.

Theory:

ASP stands for Active Server Pages. ASP is a great tool for building dynamic web pages. ASP is a program that runs inside IIS (Internet Information Services). ASP file is just the same as an HTML file. An ASP file can contain text, HTML, XML and scripts. ASP file has the file extension ñ.aspò.

ASP is Used To:

- Dynamically edit, change or add any content of a web page.
- Respond to user queries or data submitted from HTML forms.
- Access any data or databases and return the results to a browser.
- Provides security since ASP code can not be viewed from the browser.
- ASP program can minimize the network traffic.

ASP consist of six built-in objects. They are used to provide functionality to web pages. Built-in objects are included in all ASP pages. These objects enables to avoid the overhead associated with complex web programming.

The built-in objects and their description is given below:

Object	Description
Application	Stores information needed for all users of a particular application
ASPError	Reports error information. Describe the properties of the object that stores information about an error condition.
Request	Provides access to values passed to the server by the client. Describes the methods, properties and collection of the object that stores information related to the HTTP request.
Response	Controls an output from an ASP script to the requesting client. Describes the methods, properties and collections of the objects that stores information related to serverôs response.
Server	Provides access to methods and properties on the server.Withthese methods we can executecode, get error conditions, encode text strings etc.
Session	Stores information needed for a particular user session.

One of the most basic scripts we need in ASP web pages is the asp database connection script. Here are two examples of asp database connections.

This example uses a DSN connection:

```
<%
DIM objConn
Set objConn = Server.CreateObject("ADODB.Connection")
objConn.ConnectionString = "DSN=myCONNECTION.dsn"
objConn.Open
%>
```

This example uses a DSN-less connection:

```
<%
DIM objConn
Set objConn = Server.CreateObject("ADODB.Connection")
objConn.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & _
Server.MapPath ("/mydatabase.mdb") & ";"
objConn.Open
%>
```

Conclusion:

Thus the ADO connection object is used to create an open connection to a data source. Through this connection we can access and manipulate database.

EXPEIMENT NO: 10

Aim: Implement session management using ASP.

Theory:

Session Object:

Session object is used to store information about or change setting for user session. Variable store in session hold information about one single user and are available in pages in all applications.

When you are working with an application, you open it, do some change and then close it. The computer knows who you are, it knows when you start the application & when you end it. But there is one problem the web server does not know who you are & what to do because the HTTP address does not maintain the state.

ASP solves the problem by creating unique cookies. For each user the cookie is send to the client& contains information that identifies the user. This interface is called session Object.

Variables stored in session object hold information about one single user & are available to all pages in one application. Common information stored in session variable are name, id etc.

Session Object:

Collection	**Description**
Contents	Contains all the item appended to the session through a script command.
Script Object	Contains all the object append to the session with HTML<object> tag.

Session Property:

Property	**Description**
Code page	Specifies character set that will be used when displaying dynamic content
LCTD	Set or returns an integer that specifies a location or region content like date, time & currency
Session ID	Returns a unique ID for each user, the unique ID generated by server.
Timeout	Set or returns timeout period for session object in application.

Session Method:

Method	**Description**
Abandon	Destroys user session
Contents Remove	Deletes an item from contents collection
Contents	Delete all items from content collection

Remove all ()

Session Event:

Event	**Description**
Session_onend	occur when Session end
Session_onstart	occur when Session start

Conclusion:

Since HTTP is stateless protocol, the web server retains no memory of past actions and treats each browser request as a one time event. This statelessness makes it difficult for web developers to create the level of interactivity that most users expect. This can be done by creating its own session management in ASP.