Subject Name	:	WEB TECHNOLOGIES
Subject Code	:	19CS7PC01L
Department	:	**School of Computer Science and Engineering**
Program Name	:	B.Tech CSE

Prepared By:

Mrs. Charulata Palai

Mr. K L Narayana

NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY

Palur Hills, Berhampur, Odisha

Prerequisite:

- Sound knowledge in programming and Core Java.
- Basic Knowledge in Networking, Client/Server Model.

Objectives:

To provide a good understanding of the salient features of Internet and Web site design. Students are expected to understand the network concept in detail and to develop a website using serevlets and JSP.

Learning outcome and end use:

By the end of the course, the students should be able to :-

- Have a detailed understanding of the underlying design concepts of Web page.
- Be able to apply basic concept of Internet protocol and the dynamic page techniqes.
- Be able to deveop a website by using client and serverside scripting(Servletsa and JSP).

Method of Assessment:

Lab Performance	60 Marks
Record	20 Marks
Attendance	10Marks
Quiz	10 Marks
Total	100 Marks

Main principles for effective web page design

1. Include essential elements on each page.
2. Use appropriate navigational aids.
3. Keep page lengths short.
4. Use appropriate text fonts and styles.
5. Use color appropriately.
6. Keep graphics small.

For additional information concerning these six principles, please read below and check out some of the Web sites listed at the end of this page.

Six Basic Principles of Web Page Design

1. Include essential elements on each page.

Any Web page may be accessed directly from another Web site, therefore each Web page needs to contain essential information which allows it to act as an independent document. This essential information is usually placed into one of three main parts of the Web page; the header, the body, or the footer (Lynch, 1997).

The header is used to bring continuity to the various pages of the Web site, as well as indicate the main topic of the particular Web page. Therefore, the header usually contains a banner graphic which ties the various Web pages of the site together, the title of the document, and navigational aids which link to other pages within the Web site.

The body contains the main textual content of the document, as well as hypertext links to other related Web sites.

The footer is used to verify the origin and authorship of the Web page. Therefore, the footer should contain the author or contact person of the site, as well as the institution with which the author is affiliated (if any), navigational links to other pages within the Web site, the date of last revision, and a statement of copyright. Other useful information might also include the author's e-mail and mailing addresses, as well as the URL of the document.

2. Use appropriate navigational aids.

Good navigational aids are essential to good Web page design. Zimmerman (1997) points out that one way to increase the navigability of Web pages is to include "return to home page", "previous page", and "next page" links on each page. Lynch (1995), and others, suggest that local navigational links be located both at the beginning and end of the page layout so they are easily accessible to the viewer of the Web page. By including the navigational links at the end of the page, the user is not forced to return to the beginning of the Web page after browsing it in order to access another page. Another method used to increase navigability is to provide a menu or table of contents of the Web site on each Web page.

3. Keep page lengths short.

It is usually recommended that Web page lengths not exceed two or three screens worth of information (Lynch, 1995). A major disadvantage of long Web pages is that the user needs to depend on the vertical scroll bar to navigate within the page, a process which can be disorienting to the viewer. In order to keep Web pages short, longer topics can be subdivided into logical chunks of information on separate Web pages. Individual Web pages should include only relevant, yet complete, information on a single topic.

4. Use appropriate text fonts and styles.

Different web page browsers may display special non-standard text fonts in various ways. For this reason, Web page designers should use standard text fonts in designing Web pages (Vaughan, 1996). Lynch (1995) suggests using major HTML heading styles very sparingly and reducing header sizes in general so that more information can be displayed on the screen at one time without loosing legibility or causing overcrowding to the screen layout.

5. Use color appropriately.

Marcus (1990), McFarland (1995), and others point out that color should be used sparingly and only to highlight key elements of the page or to indicate specific functions. Just because you can view a particular color with a particular browser does not mean that others will be able to view it on their monitor with their browser. Therefore it is important to use browser-safe colors, which are also know as web-save colors.

Black is traditionally used on Web pages for the main body of text because of its legibility on a light background. Some colors are traditionally used to convey a particular meaning. Red, for example, is often used to signify danger or warning, and thus should be used sparingly to convey such meaning. Blue is traditionally used to indicate hypertext links to other Web pages, and a shade of purple to indicate links which have already been accessed by the current user. Altering these traditional color schemes will most likely confuse the new viewer to the Web site.

6. Keep graphics small.

Graphics can effectively be used to add interest to a web page, but the amount and size of graphics should be kept to a minimum. Too many graphics or large graphic, can take a long time to download. Usually, using several smaller graphics, as opposed to one large one, can create a better impression for visitors to your site.

SIX PRINCIPLES OF ACCESSIBLE WEB DESIGN

I. Create pages that conform to accepted standards.

II. Know the difference between structural and presentational elements; use style sheets when appropriate.

III. Use HTML 4.0 features to provide information about the purpose and function of elements.

IV. Make sure your pages can be navigated by keyboard.

V. Provide alternative methods to access non-textual content, including images, scripts, multimedia, tables, forms and frames, for user agents that do not display them.

VI. Be wary of common pitfalls that can reduce the accessibility of your site.

Assignment 1

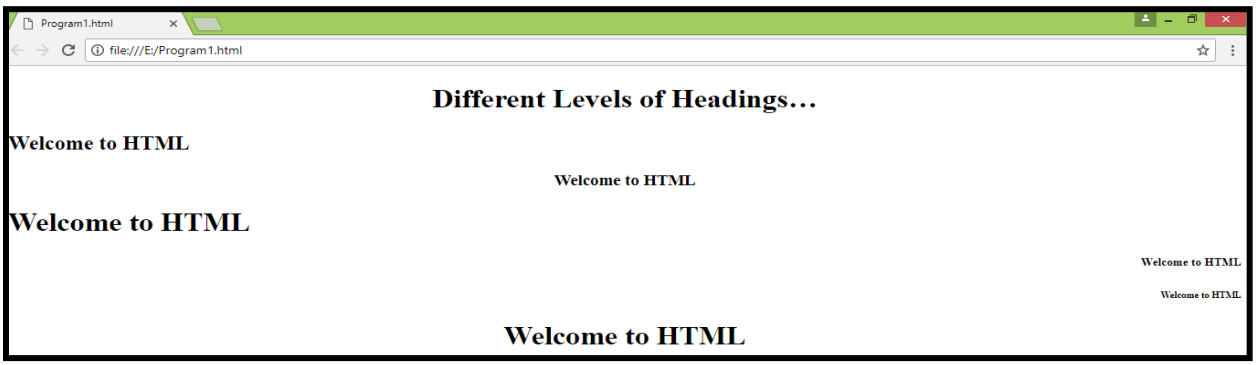
Introduction to HTML

Create a folder with your Full Roll in your Home Directory.
The web pages and all images related to the web page will be saved to that folder.

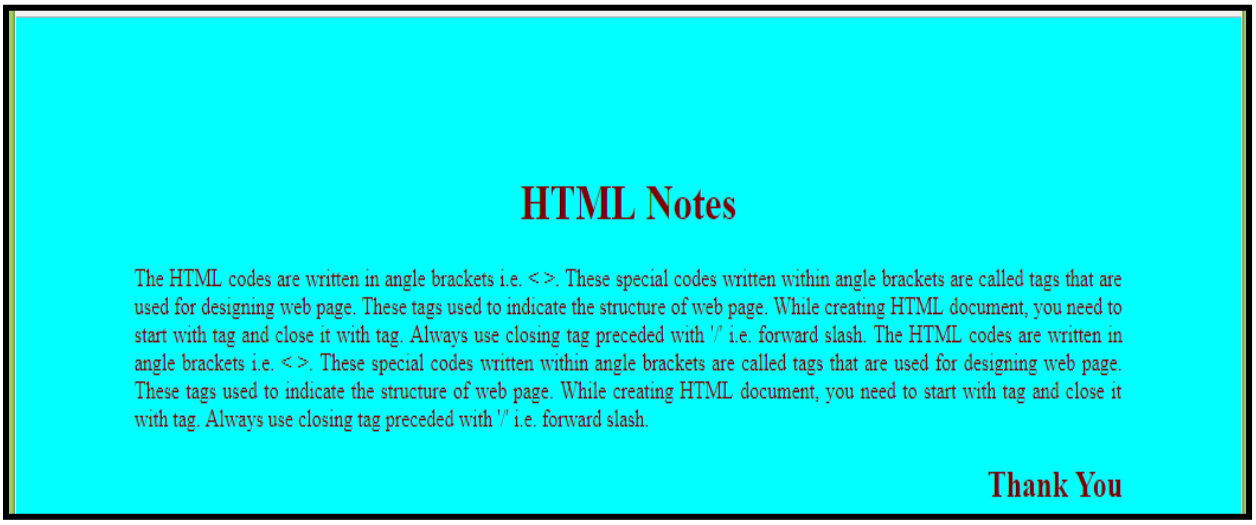
Create a page that includes the following elements:

1. Within the head section create a **web page title** (displayed in title bar of browser window): **My Favorite Things.**
2. Use a **color** for the **background** for the body.A ***centeredheading***(use the largest heading size) of: **My Favorite Things** (remember to turn off centering!).
3. Include a ***horizontal rule*** underneath the heading that is **colored**, **centered**, **75%** of the screen's width and has a **size of 8**.
4. Include the following introductory **paragraph**, filling in the blanks with the appropriate information for yourself.
5. Use a **font size of** 4 and a **font face of Comic SansMS**: My name is _____ and I am a _____ at Thurston High School. This web page lists my favorite foods, favorite television shows, and favorite movies. Remember to turn off the paragraph!
6. Insert another **blank line (a line break)** after the paragraph.
7. Using a **heading size of 2**, key the following heading: **My Favorite Foods**.
8. Using an ***unordered list***, include **five** of your favorite things to eat.
9. Use a **colored font with a size of 4**, with a font face of your choice.
10. Insert another **blank line**after the list. Using a **heading size of 2**, key the following heading: **My Favorite TV Shows**.
11. Using an ***ordered list***, list your top five favorite television shows. Use the same font settings as the unordered list. Insert another **blank line**after the list.
12. Using a **heading size of 2** again, key the following heading: **My Favorite Movies**.
13. Using a ***definition list***, list your five favorite movies and include a ***description*** of why you liked the movie. ***Underline*** the name of each movie. **Use the same font settings as the other two lists.**
14. Include ***three images*** on your page: an image representing each of your favorite things. Resize the images (using the width attribute) so that they fit appropriately on the page and are displayed in centered at the bottom of your web page (***hint:*** turn on centering before the first image tag and turn off centering after the last image tag.).
15. Create a 6 row, 6 columns Use a table border size of 1. Gives table header, insert cell values and colors. Using row span and cols pan.
16. A three column table that displays your current schedule (Period, Course Name, Teacher Name)
17. Save the page as **sample.html**

Page1.html(Different Levels of Headings)



Page2.html (paragraph)



Page3.html (Formatting)

Page4.html(Background and Inline Image)

Page5.html

Display the subject list of your Current Semester using marquee(which will scroll towards downwards)

Display the list of your 20 Friends full names, which is scrolling towards left.

Page6.html (Escape Sequence Characters)

Design the page with the following Symbols

© All Copy Rights are reserved to Mr. Shyam Sundar

® All rights are reserved to CSE Students.

½ of the Students may get O-Grade

¼ of the Students may fail if they are not perfroming well in LABS

Page7.html (Ordered List)

Apple I Phones List

I. IPHONE (2007)
II. IPHONE 3G (2008)
III. IPHONE 3GS (2009)
IV. IPHONE 4 (2010)
V. IPHONE 4S (2011)
VI. IPHONE 5 (2012)
VII. IPHONE 5C (2013)
VIII. IPHONE 5S (2013)
IX. IPHONE 6 (2014)
X. IPHONE 6 PLUS (2014)
XI. IPHONE 6S (2015)
XII. IPHONE SE (2016)
XIII. IPHONE 7 (2016)
XIV. IPHONE 7S AND IPHONE 8 OR IPHONE X (2017)

Page8.html(Ordered List)

List of Output Devices in Nested Lists

A. Monitor
  11. LED
  12. LCD
  13. CRT
B. Printers
  101. Dot Mattrix
  102. Line Printer
  103. Laser Printer
C. Plotters.
D. Projector.
E. Speaker
F. Head Phone.

Page9.html

Order List Formatting

10. This should be 10
11. This should be 11
1. This should be 1
2. This should be 2
3. This should be 3
24. This should be 24
Y. This should be Y
Z. This should be Z
AA. This should be AA
AB. This should be AB
ii. This should be ii
iii. This should be iii
iv. This should be iv

Nested List Example

A. First Item of Outermost List
B. Second Item of Outermost List
    1. First Item Of Innner List
    2. Second Item Of Innner List
        I. First Item Of Innnermost List
        II. Second Item Of Innnermost List
    3. Third Item of Innner List
C. Third Item of Outermost List

Page10.html

Unordered List Formatting

- First Item
- second Item
- third Item
- Fourth Item
- Fifth Item
- Sixth Item

Nested Unordered list

- Mega List
- Mega List
- Mega List
    - Supreme list
    - Supreme list
    - Supreme list
        - Super List
        - Super List
            - Sub List
            - Sub List
        - Super List
    - Supreme list
- Mega List
- Mega List

Assignment-2

hyperlink, Table, Frame, Form

Your web page must contain the following sections:

1) A "**Biography**" (include the word Biography as a heading) section that includes at least two paragraphs of biographical information.

2) An "**Accomplishments**" section (include the word Accomplishments as a heading) that outlines the person's major life accomplishments using an **unordered list** (include at **least five** accomplishments).

3) A "**Learn** More **About….**" (Include this as a heading) section that includes hyperlinks to **three websites** that contain additional information about your subject.

Your page needs to include:

- A title bar title within the head section
- An image (or images) using the border and ALT text attributes
- Use of heading tags (for the section headings)
- An unordered list (for the accomplishments)
- Font attributes including size, face, and color
- Hypertext links
- A page title in the title bar of Explorer
- Center your title graphic from step 1 at the top of the page.
- A colored page background.
- A colored ruled line with a width of 75% and size of 12.
- A three column table that displays your current schedule (Period, Course Name, and Teacher Name).
- A two column and 5 rows form that displays your current registration form in the website (Name,RollNumber,Stream,Branch,CollegeName)

Create a table similar to the one **below** that lists and displays pictures of five things ***you like***. The table should be 50% of the screen's width and centered between the margins.

1. Use a color for the background of your page.
2. Find five pictures from the Internet and save to the folder. Take note of what kind of image file the pictures are (jpeg, gif, bmp, etc.).
3. Create a table border size of 1.To learn how to merge two cells together.
4. Use a **heading size of 3** for the names of the items you like.

Tables

Page1.html

List of my Friends

Roll	Name	Marks
35	Ram Kumar	87
42	Diraj Panda	93
15	Kishore Kumar Prusty	52
20	Alex	67
32	St Lee	80

Page2.html

Roll No	Name	Hobbies
350	Raj Kumar Pandey	Dance, Music, Sports
142	Dipak Kumar Pattnaik	Dance, Fashion, Sports
15	Allok Kumar	Dance, Health, Sports, Movies

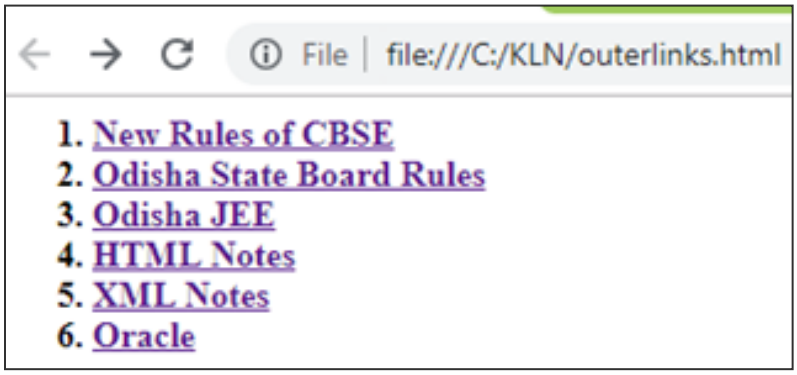
Page3.html

Student Result

Roll No	Full Name			Marks
	First	Middle	Last	
350	Raj	Kumar	Pandey	89
142	Dipak	Kumar	Pattnaik	63
15	Allok	Kumar		75

LINKING TO OUTSIDE PAGES BY USING HYPERTEXT

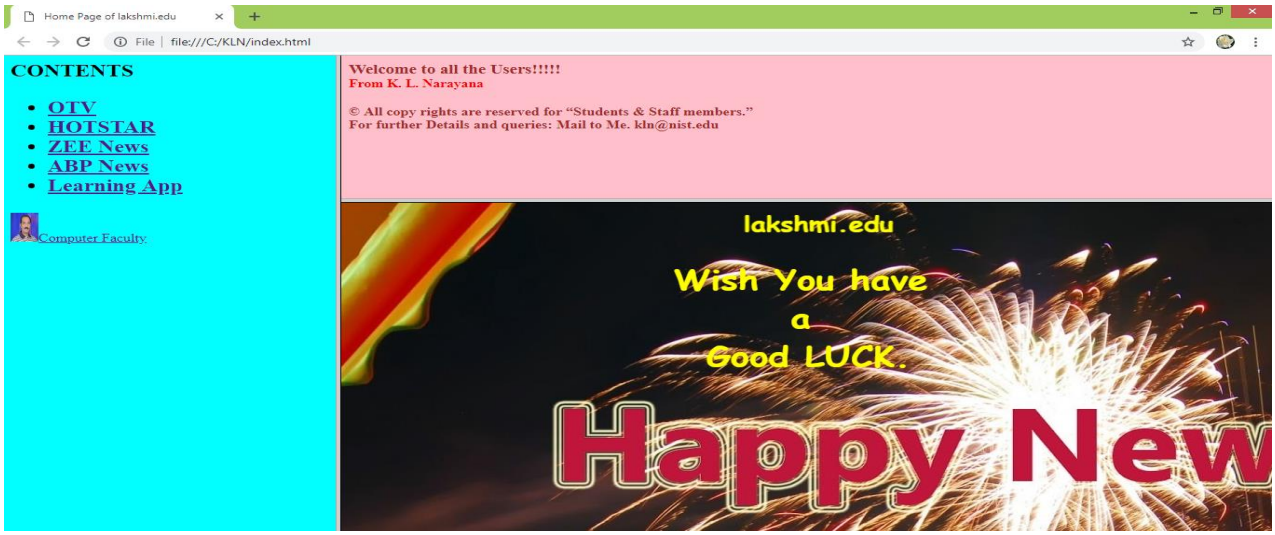
Page4.html



Frames and Hyper Links

Design the Following WEB page using frames with the help of the following 3 Pages

Contents.html	**Address.html**
	Source.html



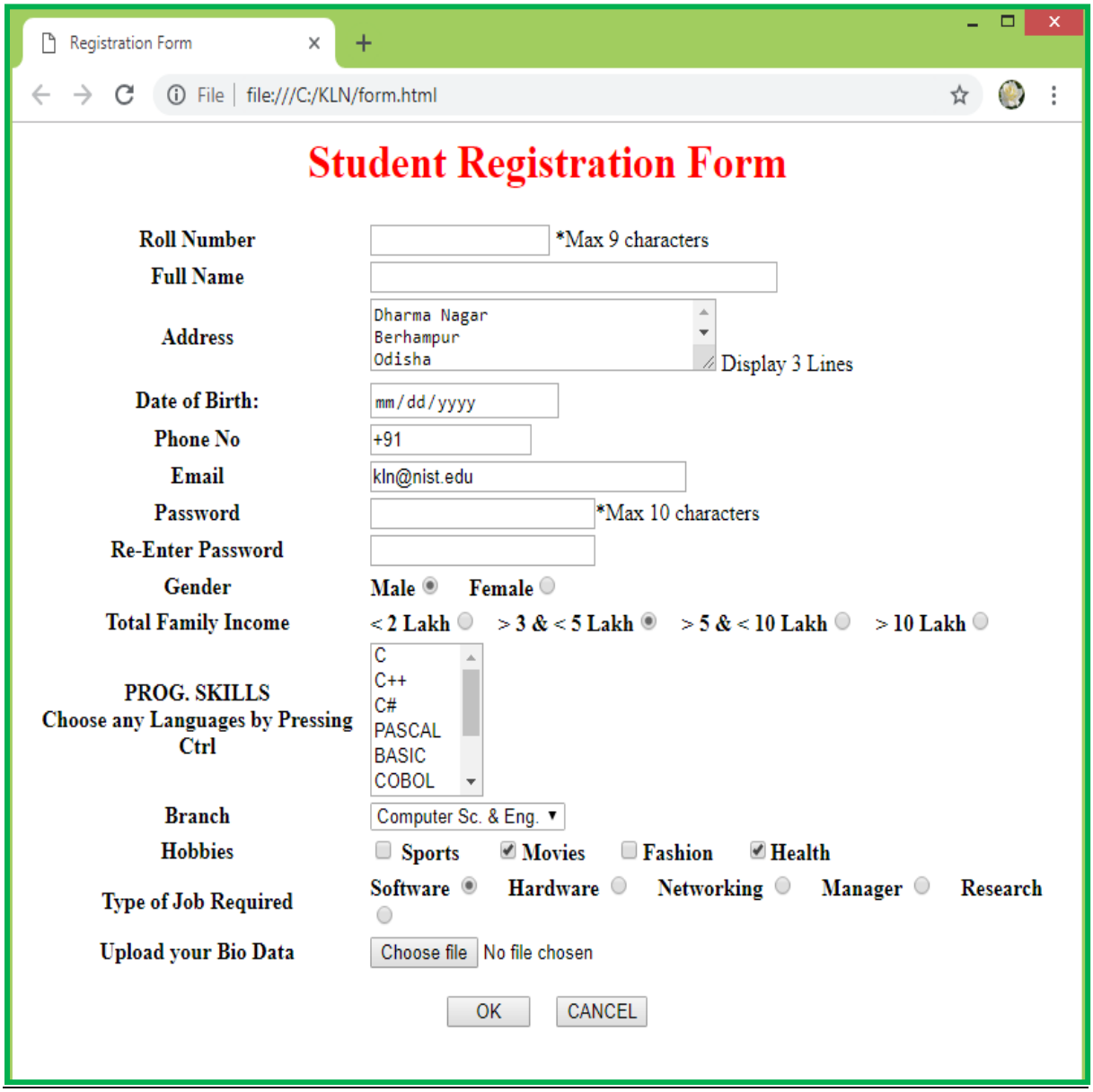
index.html

```
<html><title> Home Page of lakshmi.edu </title>
<frameset cols="25%,70%">
        <frame name="index" src="contents.html" >
        <frameset rows="30%,*">
                <frame name="address" src="address.html">
                <frame name="source" src="source.html">
        </frameset>
</frameset></html>
```

contents.html

```
<html><body bgcolor=cyan ><h2>CONTENTS<BR>
<ul>
<li><a href="https://live.odishatv.in/" target= source > OTV </a>
<li><a href="https://www.hotstar.com/channels"  target= source > HOTSTAR </a>
<li><a href="http://zeenews.india.com/live-tv"  target= source > ZEE News </a>
......
</body></html>
```

Form.html (Form Elements)



Assignment - 3(CSS)

mystyle.css

```
H1 {TEXT-ALIGN:"CENTER"; COLOR:"BLUE"}
H2 {TEXT-ALIGN:"CENTER"; COLOR:"red" ; letter-spacing:3pt}
P {     TEXT-INDENT:"+10%" ; COLOR:"maroon"; TEXT-ALIGN:"JUSTIFY";
        font-size:10pt; line-height: 3.0
  }
body {   background-color:"00ccff";  font: 9pt/1.3em verdana, arial  }
font { color="maroon" ; face:"IMPACT" }
H4  { TEXT-ALIGN:"Right"; COLOR:"navy" }
```

Question-2)

Write this page using a text editor and create more than two pages.

Here is a list of required elements you should include in these pages:

- Create the title or Logo you wish to use in Photoshop file for these pages. Titles and logos should use a nice font and one or more layer effects (such as drop shadow or outer glow). Use FILE / SAVE FOR WEB to post the title or logo to your pages as a .GIF or .JPG file.
- There should be hyperlinks on both pages that go back to your "index.html" file; back to the assignments.html page; to each other; and also include at least two links to other sites.
- The pages should each have at least two examples of font tags, for example **bold**; *italics*; or text in a different color.
- You should include at least one numbered list or one bulleted list somewhere on one or both of the pages.
- You should include at least one table somewhere on one or both of the pages.
- At least one of your images should link to another page or another site.
- You should create at least two original styles and include a style sheet (".css" document)
- At least one page should play a sound or music when the user opens it by "embedding" a sound file
- At least one page should include at least one "in-line" style

Example

```
<html><head><style>
body {
background-image: url("back1.jpg");
background-repeat: repeat-x;
```

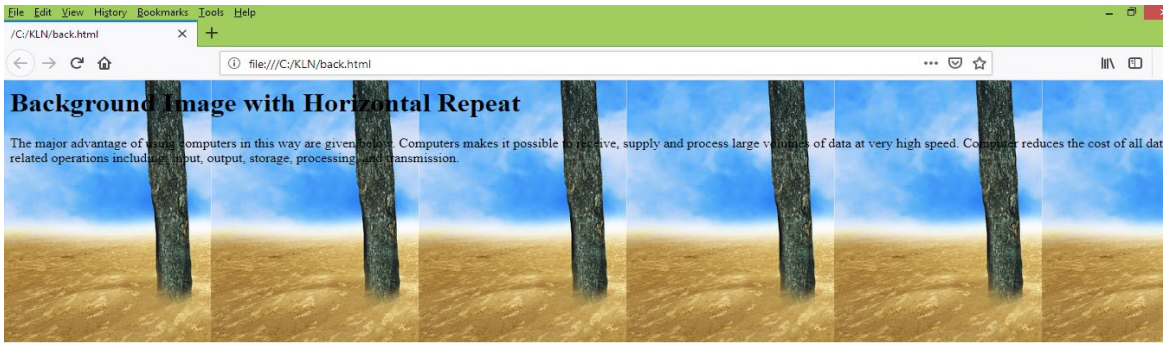
}

</style> </head> <body>

<h1> Background Image with Horizontal Repeat </h1>

 The major advantage of using computers in this way are given below. Computers makes it possible to receive, supply and process large volumes of data at very high speed. Computer reduces the cost of all data related operations including, input, output, storage, processing, and transmission.   </body></html>

Output:

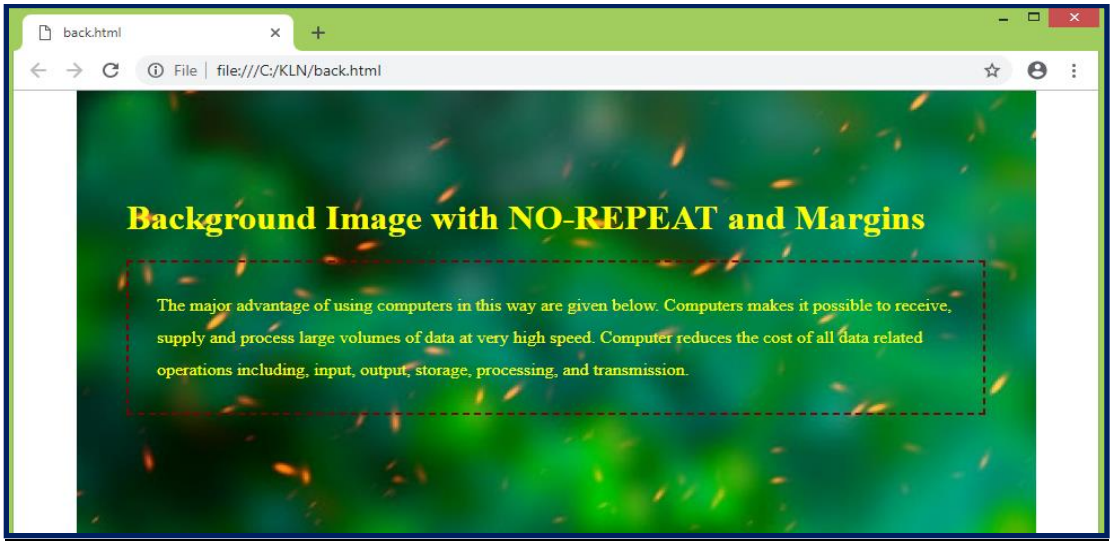


Tip: To repeat an image vertically, set background-repeat: repeat-y;

```
body
{  background-image: url("back1.jpg");
background-repeat: repeat-y;
}
```

Example (borders to the TEXT)

```
<html><head><style>
body { background-image: url("PIC.jpg");  background-repeat: no-repeat;  background-position: Center top;}
body {   margin-left:100px;  margin-top:100px;  margin-right: 100px}
body { color:yellow;  background-cyan}
p {  border-width: 2px;border-color:maroon; border-style: dashed; line-height:30px;  padding:25px }
</style><head><body><h1> Background Image with NO-REPEAT and Margins </h1>
<p  align=justify> The major advantage of using computers in this way are given below. Computers makes it possible to receive, supply and process large volumes of data at very highspeed. Computer reduces the cost of all data related operations including, input, output, storage, processing, and transmission.  </p> </body></html>
```



Example

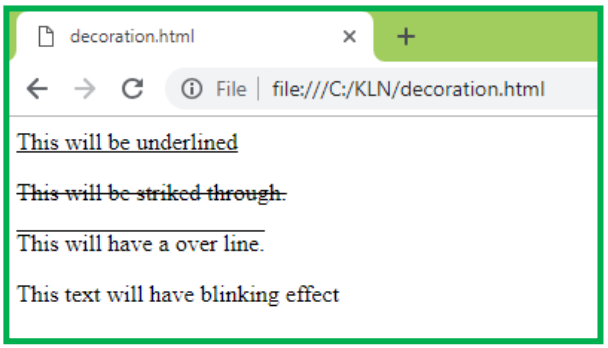
```
<html><head> <style>
.container {  position:relative;  text-align:center;  color:red;  font-size:36px;  }
.bottom-left {  position: absolute;  bottom: 8px;  left: 16px; }
.top-left {  position: absolute;  top: 8px;  left: 16px;  }
.top-right {  position: absolute;  top: 8px;  right: 16px;  }
.bottom-right {  position: absolute;  bottom: 8px;  right: 16px;  }
.centered {  position: absolute;  top: 50%;  left: 50%;  }
</style></head><body>
<h2 align=center>Image Text</h2>
<font color=blue size=7>How to place text over an image:</font>
<div class="container">
<img src="snow.jpg" alt="Snow" style="width:100%;">
<div class="bottom-left">Bottom Left</div>
<div class="top-left">Top Left</div>
<div class="top-right">Top Right</div>
<div class="bottom-right">Bottom Right</div>
<div class="centered">Centered</div>
</div> <h2>In similar way you can place any text on the top of any image in different positions. </h2>
</body></html>
```

Output:

Example

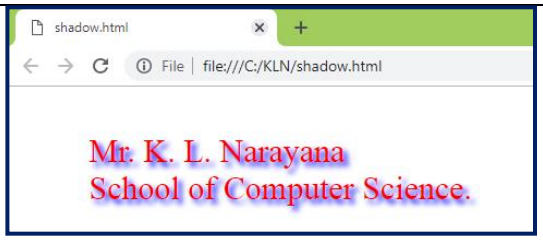
```
<html><body>
<p style = "text-decoration:underline;">
        This will be underlined
<p style = "text-decoration:line-through;">
        This will be striked through.
<p style = "text-decoration:overline;">
        This will have a over line.
<p style = "text-decoration:blink;">
        This text will have blinking effect
</body></html>
```

Output:



Example

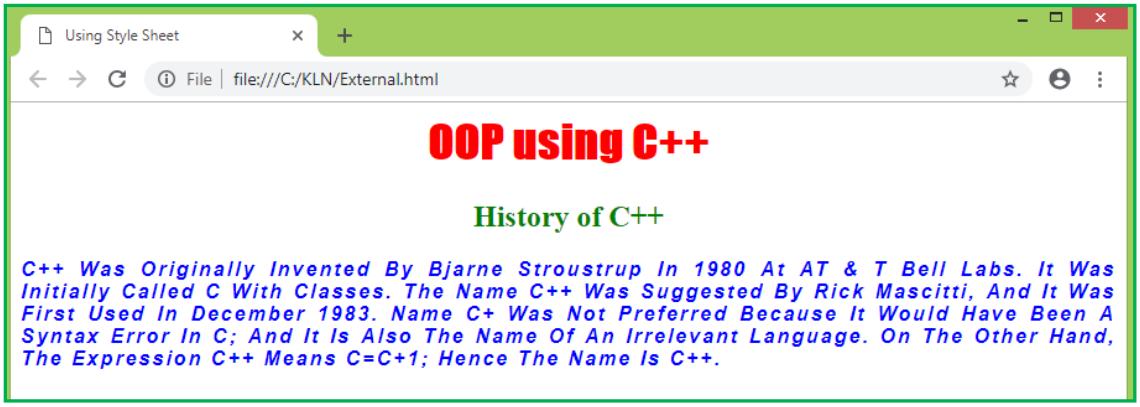
```
<html><head><style>
   p { white-space:pre ; font-size: 30px;  color:red}
p { text-shadow:4px 4px 8px  blue;>
</style></head><body>
<p>      Mr. K. L. Narayana
                School of Computer Science.
         </p>
```



</body></html>	

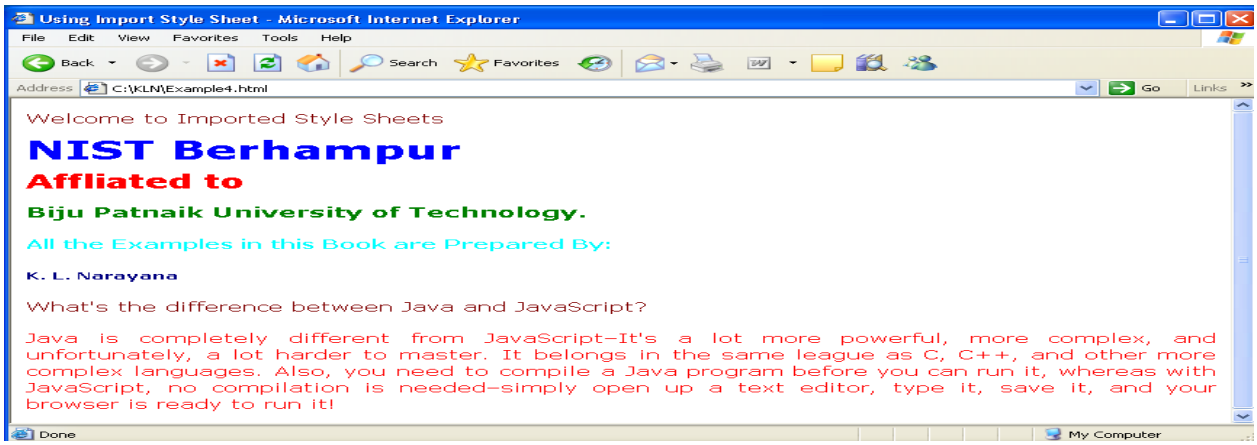
Porgram1.html

Design the following using sylte sheet tags:



Porgram2.html

Design the following using multiple sylte sheets:



Program3.html

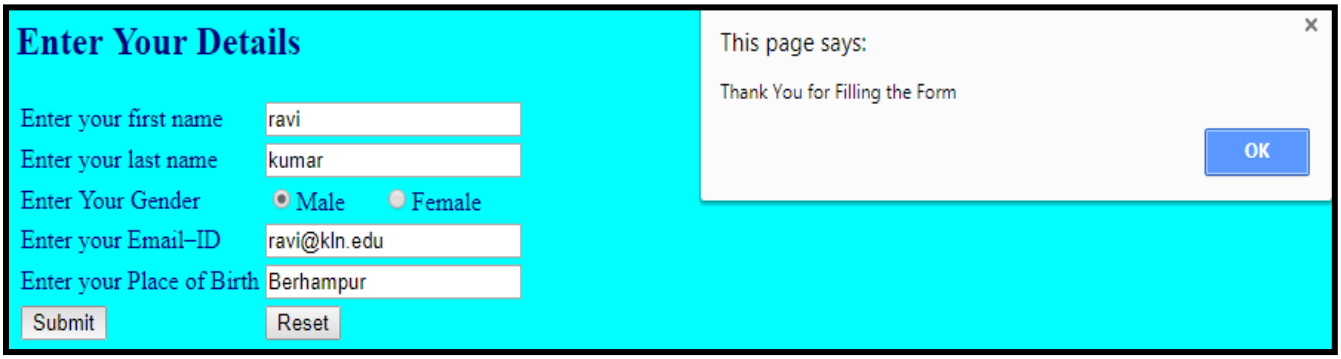
Design the BIODATA using different styles in professional way.

Assignment-4(JavaScript)

Example: Enter all the Text Fields without leaving empty which are compulsory

```
<HTML> <HEAD><TITLE>Elements Array</TITLE>
<SCRIPT LANGUAGE="JavaScript">
function verifyIt()
{	var form = document.forms[0]
	for (i = 0; i < form.elements.length; i++)
	{	if (form.elements[i].type == "text" && form.elements[i].value == "")
		{	alert("Please fill out all fields.")
			form.elements[i].focus()
			return false;
		}
	}
	alert("Thank You for Filling the Form");
}
</SCRIPT> </HEAD>
<BODY bgcolor=cyan text=navy><h2> Enter Your Details</h2>
<FORM><table cellspcing=5><tr><td>Enter your first name <td>
<INPUT TYPE="text" NAME="firstName" onKeyReleased=verifyIt()>
<tr><td>Enter your last name <td><INPUT TYPE="text" NAME="lastName">
<tr><td>Enter Your Gender<td>
<INPUT TYPE="radio" NAME="gender" checked>Male     
<INPUT TYPE="radio" NAME="gender">Female
<tr><td>Enter your Email–ID<td><INPUT TYPE="text" NAME="email">
<tr><td>Enter your Place of Birth <td><INPUT TYPE="text" NAME="city"></table>
</FORM><CENTER><FORM>
<INPUT TYPE="button" NAME="act" VALUE="Verify" onClick="verifyIt()">
</FORM></CENTER></BODY></HTML>
```

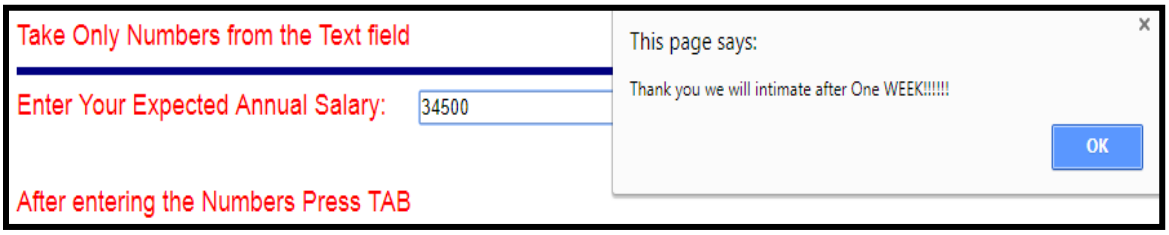
Output:



Example: Accept only digits

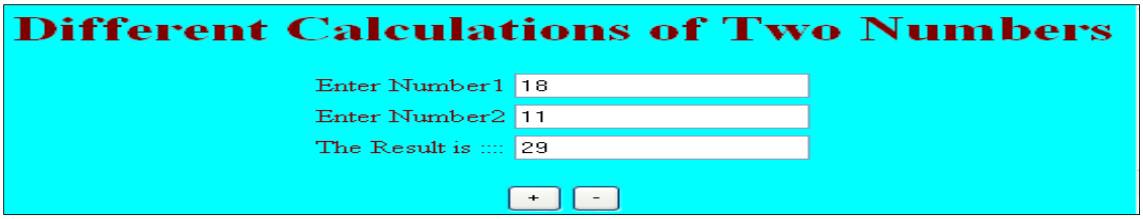
```
<HTML><HEAD><TITLE>Letting Only Numbers Pass to a Form Field</TITLE>
<SCRIPT language=JavaScript>
function checkIt(evt)
{       var charCode = (evt.which) ? evt.which : evt.keyCode
        if  (charCode < 48 || charCode > 57)
{       status = "This field accepts numbers only."
                return false
        }
        status = ""
        return true
}
</SCRIPT></HEAD>
<BODY><font color=red face=arial size=4>Take Only Numbers from the Text field
<HR size=5 color=navy><FORM onsubmit="return false">
Enter Your Expected Annual Salary:    
<INPUT onkeypress="return checkIt(event)" name=numeric
onChange='alert("Thank you we will intimate after One WEEK!!!!!!")'> Rs.
</FORM><br>After entering the Numbers Press TAB
</BODY></HTML>
```

Output:

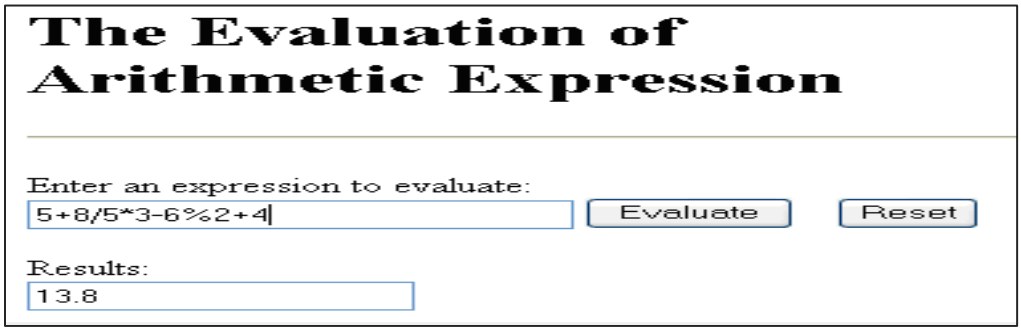


1) Write a JavaScript function that checks whether a passed string is palindrome or not?
2) Write a JavaScript function that reverses a number.
3) Write a JavaScript program which compute, the average marks of the students, then determine the Corresponding grade.
4) Write Java Script program to calculate the factorial of number given by user.
5) Write Java Script that inputs three integers from the user and outputs their sum, average, largest. Use alert dialog box to display results.
6) Program for displaying an image on mouse click and add some animation to the page.

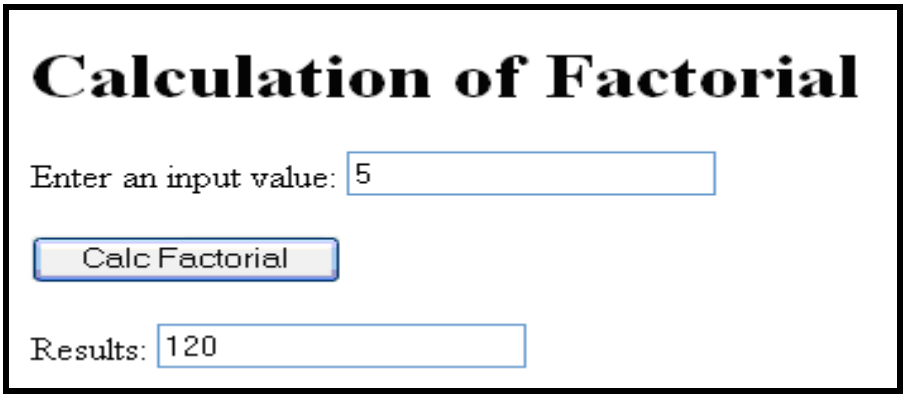
7) Design the following Web Page using JAVA Script.



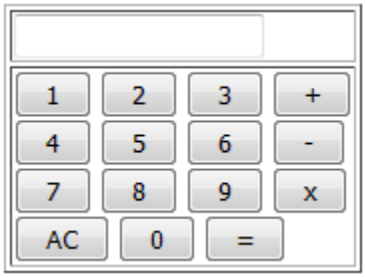
8) Design the following Web Page using JAVA Script.



9) Design the following Web Page using JAVA Script.

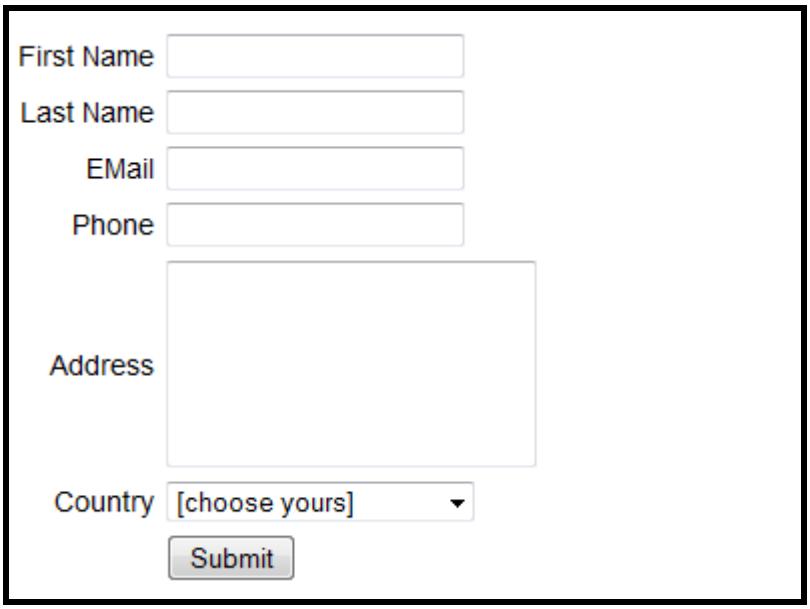


10) Design calculator of the given type using JAVA Script.



11) Create a an HTML document containing JavaScript code that

- The errors are shown in message box, one error at a time
- First Name, Last Name and Email are required fields
- First Name, Last Name max length is 20
- Phone number should be numeric only
- A Country should be selected
- Mailing address must contain @ and dot(.) symbol in it.



12) Design the following Web Page using JAVA Script

- Enter the Name (It should accept only Alphabets and Spaces)
- Enter the Phone(It should accept only Numbers)
- To display the Date of Birth
- Display the Validity date of your ATM Card

Assignment-5 (XML)

Example1.xml

```
<?xml version="1.0" ?>
<!DOCTYPE address[
<!ELEMENT address (name,college,dept,phone)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT company (#PCDATA)>
<!ELEMENT dept (#PCDATA)>
<!ELEMENT phone (#PCDATA)>
]>
<address>
<name>K L Narayana</name>
<college>NIST</college>
<dept> CSE </dept>
<phone>(0680) 249-2421</phone>
</address>
```

Output:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
- <address>
    <name>K L Narayana</name>
    <college>NIST</college>
    <dept> CSE </dept>
    <phone>(0680) 249-2421</phone>
  </address>
```

Draw the XML tree for the code given below:

```
<Office>
<FirstFloor>
<Rooms>150</Rooms>
<Lockers>200</Lockers>
</FirstFloor>
<SecondFloor>
<Rooms>225<Rooms>
<Lockers>215<Lockers>
</SecondFloor>
</Office>
```

External DTD

An external DTD is one that resides in a separate document.To use the external DTD, you need to link to it from your XML document by providing the URL of the DTD file. The URL can point to a local file using a relative reference, or a remote one (e.g, using HTTP) using an absolute reference.

Syntax

```
<!ENTITY  name  SYSTEM  "URL">
```

address.dtd

```
<!ELEMENT address(name, company, phone)>
<!ELEMENT name(#PCDATA)>
<!ELEMENT company(#PCDATA)>
<!ELEMENT phone(#PCDATA)>
```

Example.xml

```
<?xml version = "1.0" encoding = "UTF-8" standalone="yes"?>
<!DOCTYPE address SYSTEM "address.dtd">
<address>
        <name>Raj Kumar</name>
        <company>Loyala International </company>
        <phone>(011) 123-4567</phone>
</address>
```

Output:

letter.dtd

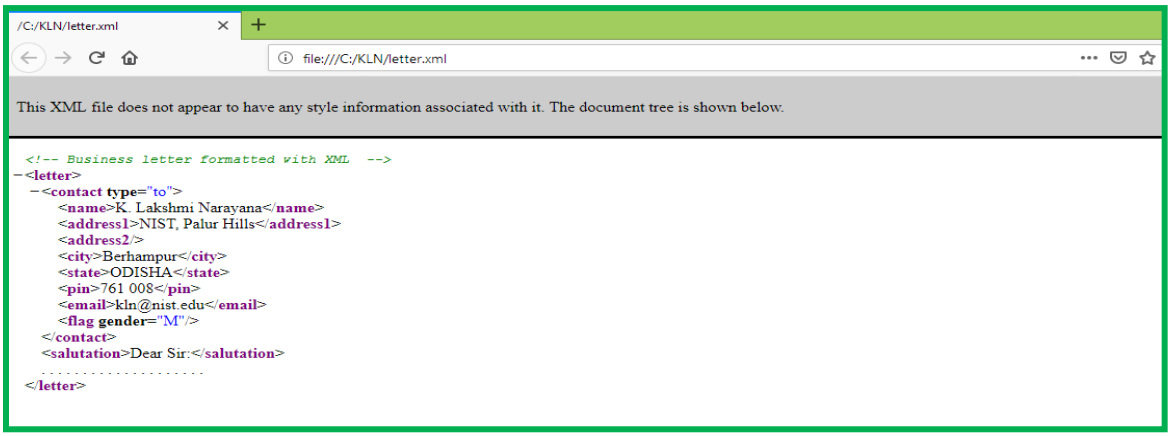
```
<!-- DTD document for letter.xml -->
<!ELEMENT letter (contact+, salutation, paragraph+, closing, signature)>
<!ELEMENT contact (name, address1, address2, city, state, pin, email, flag)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT address1 (#PCDATA)>
<!ELEMENT address2 (#PCDATA)>
<!ELEMENT city (#PCDATA)>
```

```
<!ELEMENT state (#PCDATA)>
<!ELEMENT pin (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT flag EMPTY>
<!ATTLIST flag gender (M | F) "M">
<!ELEMENT salutation (#PCDATA)>
<!ELEMENT closing (#PCDATA)>
<!ELEMENT paragraph (#PCDATA)>
<!ELEMENT signature (#PCDATA)>
```

letter.xml

```
<?xml version = "1.0"?>
<!-- Business letter formatted with XML  -->
<!DOCTYPE letter SYSTEM "letter.dtd">
<letter>
<contact type = "to">
<name>K. Lakshmi Narayana</name>
<address1>NIST, Palur Hills</address1>
<address2></address2>
<city>Berhampur</city>
<state>ODISHA</state>
<pin>761 008</pin>
<email>kln@nist.edu</email>
<flag gender = "M"/>
</contact>
<salutation>Dear Sir:</salutation>
. . . . . .
</letter>
```

Output:



note.dtd

```
<!ELEMENT note (to,from,heading,body)>
<!ELEMENT to (#PCDATA)>
<!ELEMENT from (#PCDATA)>
<!ELEMENT heading (#PCDATA)>
<!ELEMENT body (#PCDATA)>
```

If the DTD is external to your XML source file, it should be wrapped in a DOCTYPE definition with the following syntax:

note.xml

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM "note.dtd">
<note>
<to>vc@bput.ac.in</to>
<from>kln@kln.com</from>
<heading>Reg: IWT Syllabus</heading>
<body>Don&apos;t forget to modify the syllabus.</body>
</note>
```

Output:

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
– <note>
    <to>vc@bput.ac.in</to>
    <from>kln@kln.com</from>
    <heading>Reg: IWT Syllabus</heading>
    <body>Don't forget to modify the syllabus.</body>
  </note>
```

Displaying XML with CSS

Before we have learned that CSS files may work together with HTML files in the way that the former is in charge of display and the latter provides concrete information. CSS can do the same thing with XML.

catalog.css

```
CATALOG {     background-color: #ffffff;     width: 100%;     }
CD {  display: block;    margin-bottom: 30pt;    margin-left: 20;    }
TITLE {    color: “red”;    font-size: 20pt;    }
ARTIST {   color: #00FF00;    font-size: 20pt;   }
COUNTRY,PRICE,YEAR,COMPANY { display: blue;   color: #000000;   margin-left: 20pt;    }
```

catalog.xml

```
<?xml version="1.0"  ?>
<?xml-stylesheet  type="text/css" href="catalog.css"?>
<CATALOG>
<CD>
<TITLE>Empire Burlesque</TITLE>
<ARTIST>Bob Dylan</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Columbia</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1985</YEAR>
</CD>
<CD>
<TITLE>Hide your heart</TITLE>
<ARTIST>Bonnie Tyler</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>CBS Records</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1988</YEAR>
</CD>
</CATALOG>
```

Output:

Empire Burlesque	Bob Dylan	USA	Columbia	10.90	1985
Hide your heart	Bonnie Tyler	UK	CBS Records	9.90	1988

Displaying XML with HTML and JavaScript

contacts.xml

```
<?xml version="1.0" ?>
<people>
        <person>
                <name>Ivan Bayross</name>
                <address>f/12, Diamond Queen,Santacruz, Mumbai </address>
                <tel>(022)6045953</tel>
                <fax>(022)6047230</fax>
                <email>ivan@bom2.vsnl.net.in</email>
```

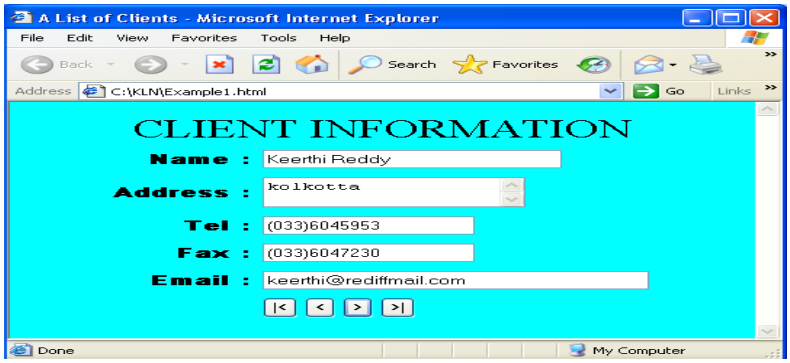
```
        </person>

        <person>
                <name>Pooja Iyenger</name>
                <address>Banglore </address>
                <tel>(080)5953930</tel>
                <fax>(080)5947061</fax>
                <email>pooja@hotmail.com</email>
        </person>
        <person>
                <name>Keerthi Reddy</name>
                <address>kolkotta </address>
                <tel>(033)6045953</tel>
                <fax>(033)6047230</fax>
                <email>keerthi@rediffmail.com</email>
        </person>

        <person>
                <name>Sishir Padhy</name>
                <address>Berhampur </address>
                <tel>(0680)205953</tel>
                <fax>(0680)207230</fax>
                <email>sishir@hotmail.com</email>
        </person>
</people>
```

contacts.html

```
<html>
<head>
<title>A List of Clients</title>
<xml id="xmlpeople" src="contacts.xml"></xml>
<script language="javascript">

function goPrev()
{       if(xmlpeople.recordset.absoluteposition > 1)
                xmlpeople.recordset.movePrevious();
}

function goNext()
```

```
{         if(xmlpeople.recordset.absoluteposition < xmlpeople.recordset.recordcount)
                  xmlpeople.recordset.moveNext();
}
</script></head>
<body bgcolor="cyan">
<CENTER>
<font face="Book Antiqua(Turkish)" font size="6">  CLIENT INFORMATION </font>
<font face="arial black">
<table width="70%" border="0" cellpadding=3 cellspacing=3 align="center">
<tr><td width="25%" align="right"><b>Name</b><td width=10% align="center">:
<td width=35%><input type="text" datasrc=#xmlpeople datafld=name size=30>
<tr><td width="25%" align="right"><b>Address</b><td width=10% align="center">:
<td width=35%><textarea row=3  datasrc=#xmlpeople  datafld=address></textarea>
<tr><td width="25%" align="right"><b>Tel</b><td width=10% align="center">:
<td width=35%><input type="text" datasrc=#xmlpeople datafld=tel>
<tr><td width="25%" align="right"><b>Fax</b><td width=10% align="center">:
<td width=35%><input type="text" datasrc=#xmlpeople datafld=fax>
<tr><td width="25%" align="right"><b>Email</b><td width=10% align="center">:
<td width=55%><input type="text" datasrc=#xmlpeople datafld=email size=40>
<tr><td width="25%"><td width="10%">
<td width="35%">
<input id="first" type="button" value=" |<"  onClick="xmlpeople.recordset.moveFirst()">
<input id="prev" type="button"  value=" < " onClick="goPrev()">
<input id="next" type="button"  value=" > " onClick="goNext()">
<input id="last" type="button"  value=" >| " onClick="xmlpeople.recordset.moveLast()">
</table></font>
</CENTER></body></html>
```

Output:



Example:

```
<HTML> <HEAD> <TITLE> Dances of India </TITLE> </READ>
```

```
<body  BGCOLOR="yellow" LINK="green" ALINK= "blue"  VLINK="RED">
<Hl ALIGN="CENTER">
<center><FONT COLOR="RED" size=7>  Dances of India </FONT></center>
</Hl> <IMG SRC ="dance.jpg" ALIGN="right"  width=120 height=150>

<P align=justify>
```

There are many types of dance in India, from those' which are deeply religious in content to those which are danced on more trivial happy occasions. Classical dances of India are usually always spiritual in content, although this is often true also of folk dances. There are many types of dance in India, from those' which are deeply religious in content to those which are danced on more trivial happy occasions. Classical dances of India are usually always spiritual in content, although this is often true also of folk dances. There are many types of dance in India, from those' which are deeply religious in content to those which are danced on more trivial happy occasions. Classical dances of India are usually always spiritual in content, although this is often true also of folk dances.

```
<TABLE ALIGN="CENTER"BORDER = 2 BORDERCOLOR ="BLUE" BGCOLOR=slate>
<CAPTION>
<FONT COLOR = "BLUE" > DANCES OF INDIA </FONT>
</CAPTION>
<TR>
<TH ALIGN="CENTER"> CLASSICAL DANCES
<TH ALIGN="CENTER"> FOLK DANCES
<TR>
<TD ALIGN="LEFT"> Mahini AttaIn
<TD ALIGN="LEFT"> Hikat of Himachal Pradesh
<TR>
<TD ALIGN="LEFT"> Bharata Natyam
<TD ALIGN="LEFT"> Rouff of Kashmir
<TR>
<TD ALIGN="LEFT"> Odissi
<TD ALIGN="LEFT"> Hurkia Baol of Kumaon
</TABLE>
<BR>
List
<BR><A href ="one.html">Mohini Attam</a>
<BR><A href ="two.htmL">Bharats Natyam </a>
<BR><A href ="three.html"> Odissi <fa>
<BR><A href ="two .html"> Bharata Natyam </a>
</BODY> </HTML>
```

Question -1:

Write an application to create a XML document from a university employee database .The XML document should contain the following:

i) Employee code

ii) Employee Name

iii) Designation

iv) Address

v) Department

vi) The last twelve month performance summary

Questions-2 - You will build two XML files, one for a nested model and one for an empty model, and email them to me as attachments. Call them **filename_nested.xml or address_book_nested.xml, and filename_empty.xml or address_book_empty.xml, or whatever (use your model name).**

Questions-3 **-** You will create a DTD for the nested and empty model files in questions and assignment one, and the linking files to them. Call them **filename_nested_dtd.xml, filename_nested.dtd, filename_empty_dtd.xml, and filename_empty.dtd (using your 'filename').**

Questions-4 - Write the HTML code to generate the following output of a table with content exactly in the same format as shown within the table:

Marks			
English	**Hindi**	**Maths**	**Science**

Questions-5 - Write an application to create a XML document.

```
  <?xml version="1.0" ?>
- <people>
  + <person>
  - <person>
      <name>Santosh</name>
      <address>Industrial, Berhampur</address>
      <tel>(0680)2492422</tel>
      <fax>(0680)2492627</fax>
      <email>santosh@hotmail.com</email>
    </person>
  - <person>
      <name>nagaraju achary</name>
      <address>kolkotta</address>
      <tel>(033)6045953</tel>
      <fax>(033)6047230</fax>
      <email>nagagraj@rediffmail.com</email>
    </person>
  - <person>
      <name>NIST</name>
      <address>Golanthara</address>
      <tel>(0680)6045953</tel>
      <fax>(0680)6047230</fax>
      <email>nist@nist.edu</email>
    </person>
  </people>
```

Questions-6 - Write the HTML code to generate the following web page with the given below specifications:

(a) Bordered table should have background color in pink.

(b) Table's header row with a heading "INCOME TAX SLABS 2017-18" should spread over four cells.

(c) After heading row, first cell of next row should spread over five rows with an image named"it.jpg" stored in d: drive.

(d) Set the space between the cell wall and the cell content to 10 pixels and set the space between the cells to 10 pixels.

(e) At the bottom of the page, a link to next page is there which is linked to another webpage named "next.html".

INCOME TAX SLABS 2017-18			
	S.No.	Income Range	Tax%
	1	0 – 250000	NIL
	2	250001 - 500000	10%
	3	500001 - 1000000	20%
	4	>1000000	30%

Next Page

Assignment-6
HTML5 and CSS3

GEOLOCATION in HTML5

Geolocation allows you to give your current location information to the browser.

Current Location

Using JavaScript, we can gather the details of our current location, or whichever device we are on.

Google Maps

Using the Google Maps API v3, we can create a Google Map on the fly and add our current position to it.

Page1.html:Show your current location using google maps.

```
<!DOCTYPE html><html lang="en"><head><meta charset="utf-8">
<title>HTML5 Geolocation</title>
<script src="https://maps.google.com/maps/api/js?sensor=false"></script>
<script type="text/javascript">
function showPosition(){
if(navigator.geolocation)
navigator.geolocation.getCurrentPosition(showMap, showError);
else    alert("Sorry, your browser does not support HTML5 geolocation.");
}

// Define callback function for successful attempt
function showMap(position){
   // Get location data
lat = position.coords.latitude;
long = position.coords.longitude;
var latlong = new google.maps.LatLng(lat, long);
var myOptions = {
center: latlong,
zoom: 16,
mapTypeControl: true,
navigationControlOptions: {style:google.maps.NavigationControlStyle.SMALL}
   }
var map = new google.maps.Map(document.getElementById("embedMap"), myOptions);
var marker = new google.maps.Marker({position:latlong, map:map, title:"You are here!"});
}
```

```
// Define callback function for failed attempt
function showError(error){
if(error.code == 1)
    result.innerHTML = "You've decided not to share your position. We won't  ask you again.";
else if(error.code == 2)
       result.innerHTML = "The network is down or the positioning service can't be reached.";
else if(error.code == 3)
       result.innerHTML = "The attempt timed out before it could get the location data.";
else
       result.innerHTML = "Geolocation failed due to unknown error.";
}
</script></head><body>
<button type="button" onclick="showPosition();">Show My Position on Google Map</button>
<div id="embedMap" style="width: 400px; height: 300px;">
<!--Google map will be embedded here-->
</div></body></html>
```

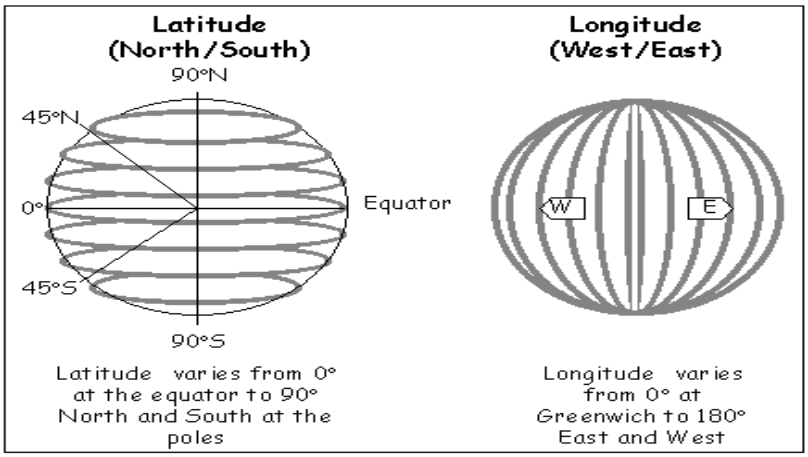
Output:

Latitude and Longitude

Latitude and longitude are imaginary (unreal) lines drawn on maps to easily locate places on the Earth. Latitude is distance north or south of the equator (an imaginary circle around the Earth halfway between the North Pole and the South Pole) and longitude is distance east or west of the prime meridian (an imaginary line running from north to south through Greenwich, England). Both are measured in terms of the 360 degrees (symbolized by °) of a circle.

Latitude is an angle (defined below) which ranges from 0° at the Equator to 90° (North or South) at the poles. Lines of constant **latitude**, or parallels, run east–west as circles parallel to the equator. **Latitude** is used together with longitude to specify the precise location of features on the surface of the Earth.

Longitude is measured in degrees east or west of the prime meridian. This means one half of the world is measured in degrees of east longitude up to 180°, and the other half in degrees of west longitude up to 180°. See the diagrams below to understand latitudes and longitudes better.



Page2.html:

```
<!DOCTYPE html>
<html><body>
<p>Click the button to get your coordinates.</p>
<button onclick="getLocation()">Try It</button>
<p id="demo"></p>
<script>
var x = document.getElementById("demo");
function getLocation() {
   if (navigator.geolocation) {
      navigator.geolocation.getCurrentPosition(showPosition);
   } else {
      x.innerHTML = "Geolocation is not supported by this browser.";
   }
}

function showPosition(position) {
   x.innerHTML = "Latitude: " + position.coords.latitude +
   "<br>Longitude: " + position.coords.longitude;
}
</script></body></html>
```

Output :

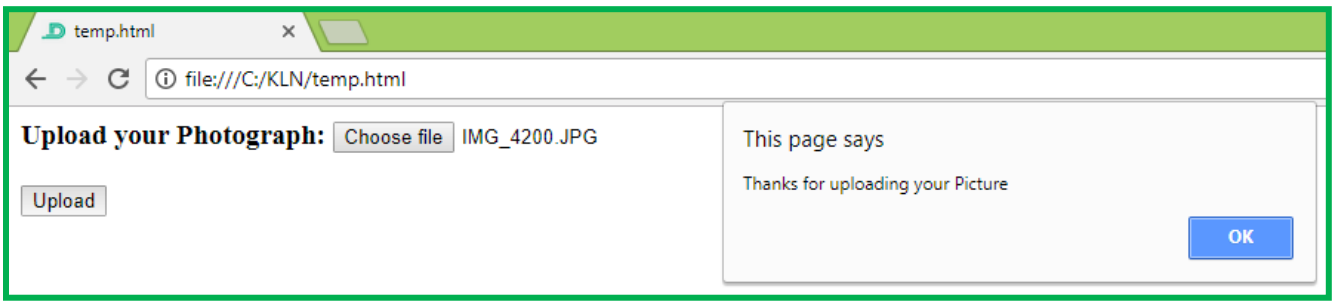
Click the button to get your coordinates.

Try It

Latitude: 20.2743
Longitude: 85.8375

Regular Expression for allowing Image files only

```
<form action="" method="post">
<script>
function ValidateExtension() {
var allowedFiles = [".png", ".jpg", ".gif"];
var fileUpload = document.getElementById("fileUpload");
var lblError = document.getElementById("lblError");
var regex = new RegExp("([a-zA-Z0-9\s_\\.\-:])+(.png|.jpg|.gif)$");
if (!regex.test(fileUpload.value.toLowerCase())) {
        lblError.innerHTML = "Please upload file extensions: <b>" + allowedFiles.join(', ') + "</b> only.";
return false;
      }
else
{ alert("Thanks for uploading your Picture");
return true;
   }
   }
</script>
<h3> Upload your Photograph:
<input id="fileUpload" type="file" />
<br />
<span id="lblError" style="color: red;"></span>
<br />
<input type="submit" id="btnUpload" value="Upload" onclick="return ValidateExtension()" />
</form>
```

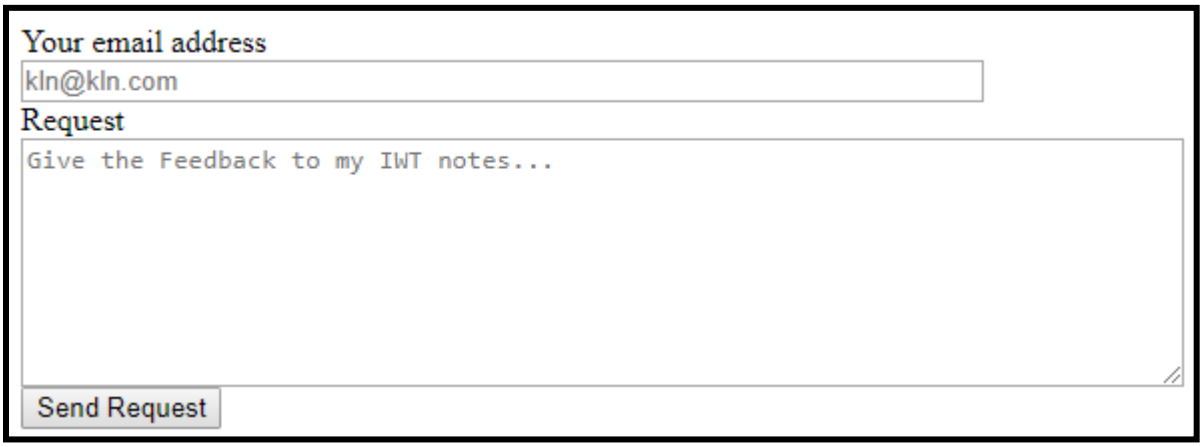


Q1) Design the Follwing Page:

Q2) Design the Follwing Page:

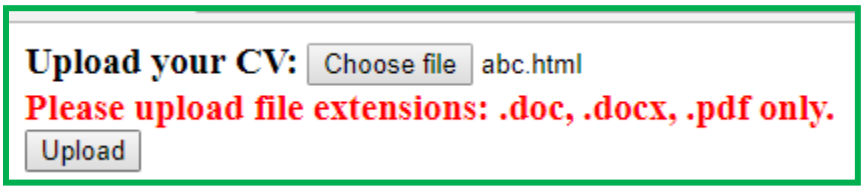
Q3) Design the Follwing Page:

Email-Validations

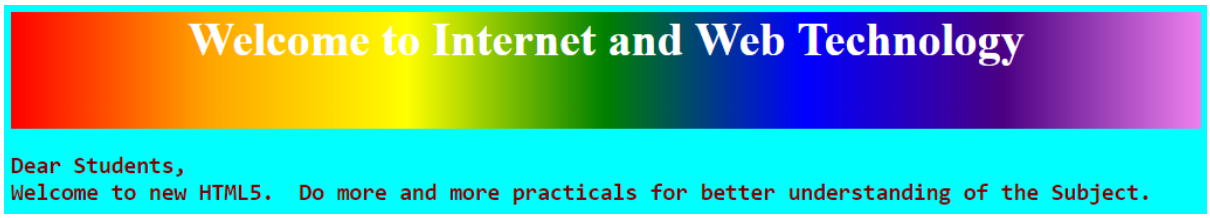


Q4) Design the Follwing Page:

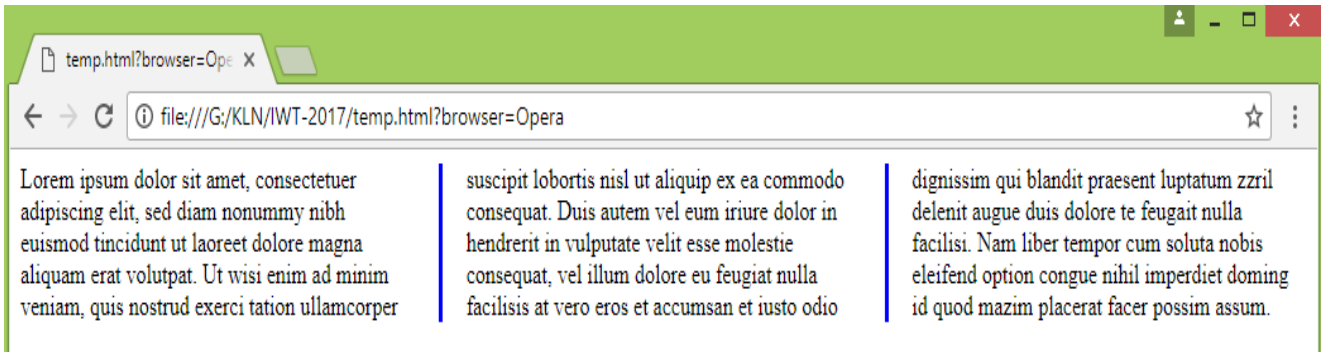
Regular Expression for allowing Word Document and PDF files only with size 100 kb to 1 mb



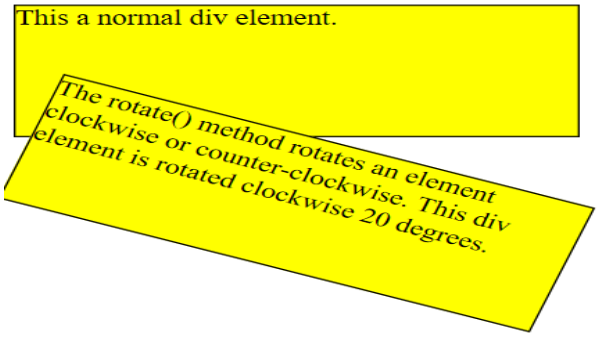
Q5) Design the Follwing Page:



Q6) Design the Follwing Page:



Q7) Design the Follwing Page:



Assignment -07
Servlets Basics

What is Servelet:

- A *servlet* is a Java programming language class that is used to extend the capabilities of servers that host applications accessed by means of a request-response programming model.
- A servlet is a technology used to create web application which acts as an intermediary between the client and the server.
- Servlet is a web component that is deployed on the server to create dynamic web page.
- As servlet modules run on the server, they can receive and respond to requests made by the client.
- Share data with other Servlets, making useful things like database connection pools easy to implement.

Servlet Container

A web browser will send the request in HTTP format. The Servlet container will convert that into a request object. Similarly the response object - populated by the Servlet is converted into an HTTP response by the Servlet container.

Using Servlets, you can collect input from users through web page forms, present records from a database, and create web pages dynamically.

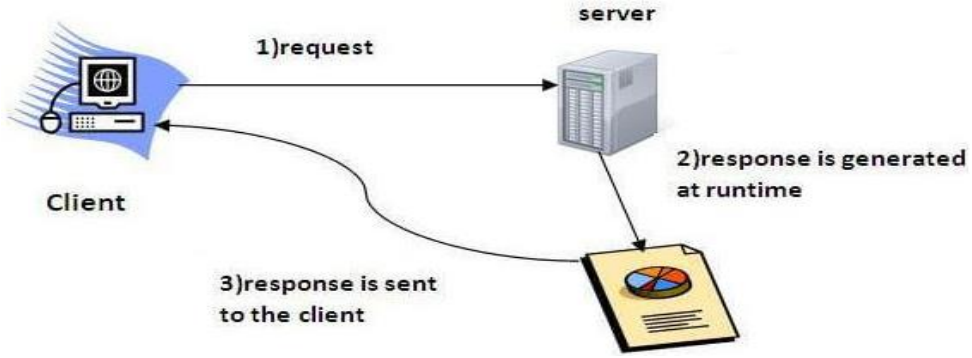


Fig: Servlet Container

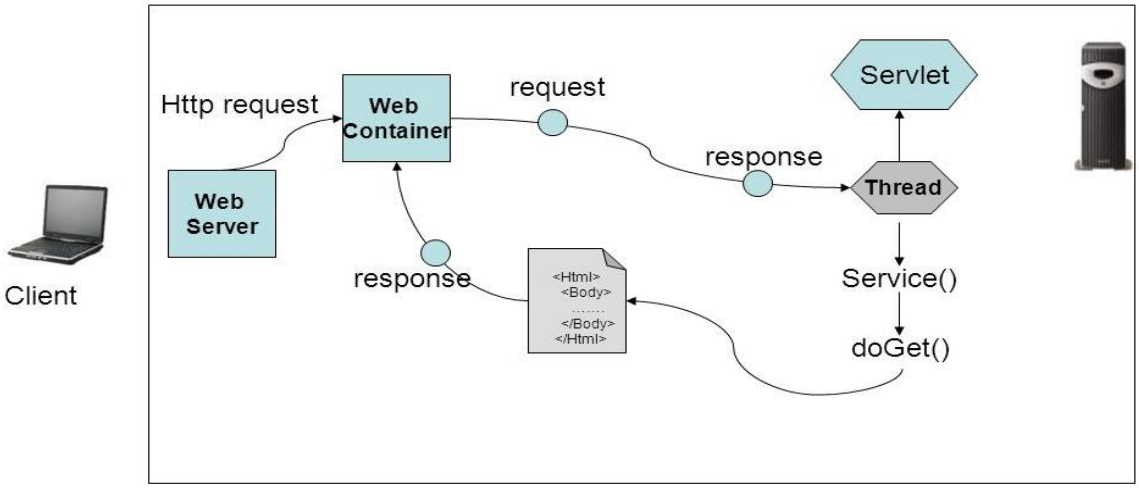


Fig: Container Handles request and response

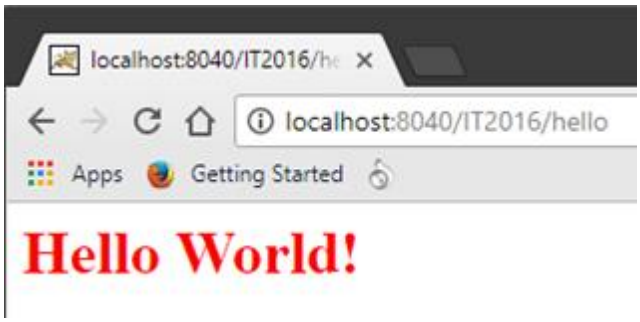
HelloWorld.java

```
import java.io.*;
import javax.servlet.*;
 import javax.servlet.http.*;
 public class HelloWorld extends HttpServlet {
        protected void doGet(HttpServletRequest req, HttpServletResponse res)
        throws ServletException, IOException {
                res.setContentType("text/html");
                PrintWriter out = res.getWriter();
                out.println("<html>");  out.println("<body>");
                out.println("<h1><font color='red'>Hello World!</font></h1>");
                out.println("</body> </html>");
                out.close();
         }
 }
```

web.xml

```
<web-app>
        <servlet>
                <servlet-name>helloprog</servlet-name>
                <servlet-class>HelloWorld</servlet-class>
                </servlet>
        <servlet-mapping>
                <servlet-name>helloprog</servlet-name>
                <url-pattern>/hello</url-pattern>
        </servlet-mapping>
</web-app>
```

Output:



Sending form data to servlet:

Initially create a form say form.html and store it in the folder:
C:\Program Files (x86)\Apache Software Foundation\Tomcat 8.5\webapps\IT2019

from.html

```
<html>
        <form action="/IT2017/ois/index" method="GET">
Enter name <input type="text" name="uname">
                <input type="Submit" value="Submit">
                <input type="Reset" value="Cancel">
        </form>
</html>
```

ParameterPassing.java

```
/* In this code we are accepting name and displaying a Hello message along with the name */
import javax.servlet.*;
import javax.servlet.http.*;
import java.io.*;
public class ParameterPassing extends HttpServlet
{       public          void          service(HttpServletRequest          req,HttpServletResponse          res)throws
        ServletException,IOException
        {       String uname=req.getParameter("uname");
                res.setContentType("text/html");
                PrintWriter out=res.getWriter();
                out.println("<html>");
                out.println("<body>");
                out.println("<h1>Hello "+uname+"<br>");
                out.println("Welcome to Our Web Site.</h1></body>");
                out.println("</html>");
        }
}
```

web.xml

```
<web-app>
<servlet>
   <servlet-name>helloprog</servlet-name>
   <servlet-class>HelloWorld</servlet-class>
   <servlet-name>pp</servlet-name>
   <servlet-class>ParameterPassing</servlet-class>
```

```
  </servlet>
   <servlet-mapping>
        <servlet-name>helloprog</servlet-name>
        <url-pattern>/hello</url-pattern>
        <servlet-name>pp</servlet-name>
     <url-pattern>/ois/index</url-pattern>
  </servlet-mapping>
</web-app>
```

1. Write a JSP program to check a number is even or odd.
2. Write a JSP program to check whether a number is an Armstrong number.
3. Create a registration form and display the values entered by the user in another page using servlets.
4. Write a servlet program to create HttpServlet and display your name.
5. Write a servlet program to display the number from 0 to 10.
6. Write a servlet program to perform arithmetic operations between two numbers.
7. Write a servlet code to pass the following Student data and print the same in Web Server.
   Roll,Name,Branch,Gender
8. Write a servlet code to pass the following Student roll, name and 3 subject's marks and process the result as per the University Rule.

Cookies

- *getCookies*

The getCookies method returns the contents of the Cookie header, parsed and stored in anarray of Cookie objects. This method is discussed in more detail in Chapter 8 (HandlingCookies).

- *getAuthTypeandgetRemoteUser*

ThegetAuthType andgetRemoteUsermethodsbreaktheAuthorization header into itscomponentpieces.

- *getContentLength*

ThegetContentLengthmethodreturnsthevalueoftheContent-Length header (as anint).

- *getContentType*

ThegetContentTypemethodreturnsthevalueoftheContent-Typeheader(asa String).

- *getDateHeaderandgetIntHeader*

The getDateHeader and getIntHeader methods read the specified headers and then convertthemtoDateandintvalues,respectively.

- *getHeaderName*

Rather than looking up one particular header, you can use the getHeaderNames method togetanEnumeration ofallheadernamesreceivedonthisparticularrequest.

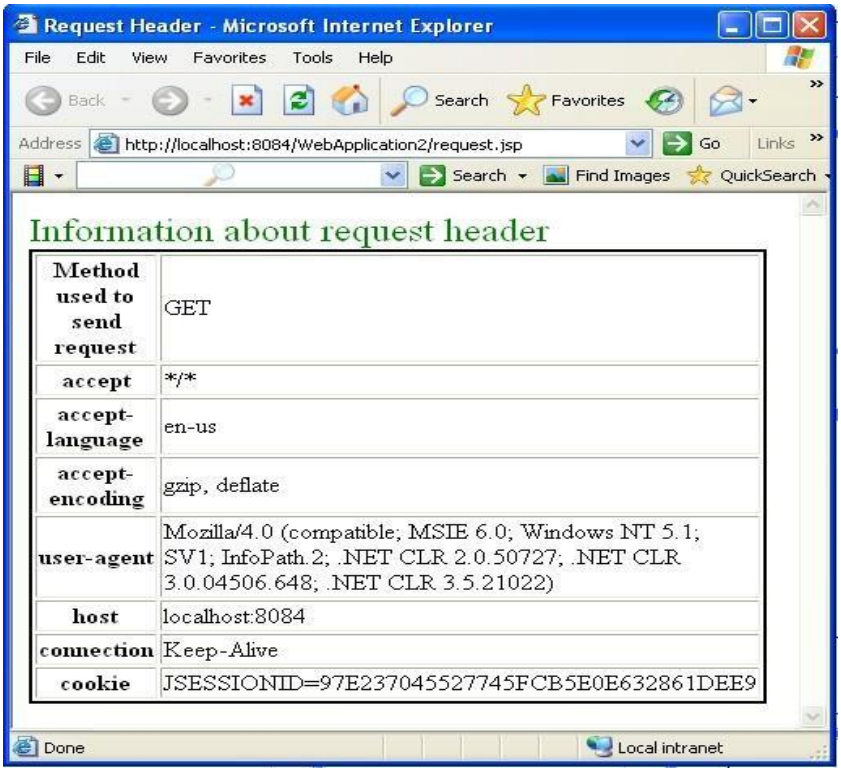
getHeaders
Inmostcases,eachheadernameappearsonly onceintherequest.Occasionally,however,a header can appear multiple times, with each occurrence listing a separate value. Accept-Language is one suchexample. You can use getHeaderstoobtainanEnumerationofthevaluesofalloccurrencesoftheheader.

Program:

```
<%@pagecontentType="text/html"pageEncoding="UTF-8"%>

<!DOCTYPEhtml>

<html>
<head>
<metahttp-equiv="Content-Type"content="text/html;charset=UTF-8">
<title>RequestHeader</title>
</head>

<body>
<fontsize="+2"color="green">Informationaboutrequestheader</font>
<br>
<tablestyle="background-color:white;border-color:black;width:50%"border="2">
<tr>
<th>Methodusedtosendrequest</th>
<td><%=request.getMethod()%></td>
</tr>

<% java.util.Enumerationnames=request.getHeaderNames();while(names.hasMoreElements())
{
Stringhname=(String)names.nextElement();
%>
<tr>
<th><%=hname%></th>
<td><%=request.getHeader(hname)%></td>
</tr>
<%
}
%>
```

```
</table>
</body>
</html>
```

Output



1. Servlet_contextIndex.html

```
<html><body>
<fontface="verdana"size="2px">
<formaction="onContext"method="post">Exampleon ServletContext<br>
<inputtype="submit"value="ClickHere">
</form></font></body></html>
```

Context.java

```
importjava.io.IOException;importjava.io.PrintWriter;
import javax.servlet.ServletContext;import javax.servlet.ServletException;importjavax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;importjavax.servlet.http.HttpServletResponse;

publicclassContextextendsHttpServlet
{
protectedvoiddoPost(HttpServletRequestreq,HttpServletResponseres)throwsServletException,IOException
{
```

```
PrintWriter pw=res.getWriter();res.setContentType("text/html");ServletContext
context=getServletContext();String s1=context.getInitParameter("n1");Strings2=context.getInitParameter("n2");
pw.println("n1valueis"+s1+"andn2is"+s2);pw.close();
}
}
```

Web.xml

```
<web-app>
<context-param>
<param-name>n1</param-name>
<param-value>100 </param-value>
</context-param>
<context-param>
<param-name>n2</param-name>
<param-value>200 </param-value>
</context-param>
<servlet>
<servlet-name>sc</servlet-name>
<servlet-class>Context</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>sc</servlet-name>
<url-pattern>/onContext</url-pattern>
</servlet-mapping>
<welcome-file-list>
<welcome-file>index.html</welcome-file>
</welcome-file-list>
</web-app>
```

2. Request Dispatcher

Index.html

```
<html><body>
<formaction="loginPage"method="post">
User Name:<input type="text" name="uname"/><br/>Password:<inputtype="password"name="upass"/><br/>
<inputtype="submit"value="SUBMIT"/>
</body>
</form>
</html>
```

WelcomeUser

```
import java.io.*;importjavax.servlet.*;
importjavax.servlet.http.*;

publicclassWelcomeUserextendsHttpServlet{

publicvoiddoPost(HttpServletRequestrequest,HttpServletResponseresponse)
throwsServletException,IOException
{

response.setContentType("text/html");PrintWriterpwriter=response.getWriter();

String name=request.getParameter("uname");pwriter.print("Hello"+name+"!");pwriter.print("Welcome to nist");
}

}
```

Validation.java

```
import java.io.*;import java.io.*;importjavax.servlet.*;
importjavax.servlet.http.*;
publicclassValidationextendsHttpServlet
{
publicvoiddoPost(HttpServletRequestrequest,HttpServletResponseresponse)
throwsServletException,IOException
{
response.setContentType("text/html");
PrintWriter pwriter = response.getWriter();String name=request.getParameter("uname");String pass=request.getParameter("upass");if(name.equals("nist") &&
pass.equals("nistnist"))
{
RequestDispatcherdis=request.getRequestDispatcher("welcome");dis.forward(request,response);
}
else
{
pwriter.print("User name or password is incorrect!");RequestDispatcherdis=request.getRequestDispatcher("index.html");dis.include(request,response);
```

```
}
}
}
```

Web.xml

```
<web-app>

<servlet>
<servlet-name>Login</servlet-name>
<servlet-class>Validation</servlet-class>
</servlet>

<servlet>
<servlet-name>Welcome</servlet-name>
<servlet-class>WelcomeUser</servlet-class>
</servlet>

<servlet-mapping>
<servlet-name>Login</servlet-name>
<url-pattern>/loginPage</url-pattern>
</servlet-mapping>

<servlet-mapping>
<servlet-name>Welcome</servlet-name>
<url-pattern>/welcome</url-pattern>
</servlet-mapping>

</web-app>
```

1. Write a program that creates an HTTP servlet to perform session tracking.
2. Create an HTTP servlet to perform various database operations.

Assignment-08
Database operations in web application

JDBC Example: CREATE and INSERT

```
import java.sql.*;
class OracleCon{
  public static void main(String args[]){
  try{      Class.forName("oracle.jdbc.driver.OracleDriver");
    Connection con=
DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:xe","system","oracle");
    Statement stmt=con.createStatement();
    stmt.execute("create table student_555(name varchar2(20),roll number(10))");
    stmt.execute("insert into student_555  values('ram',123)");
    stmt.execute("insert into student_555  values('shyam',125)");
    con.close();
  }
   catch(Exception e){ System.out.println(e);}
  }
}
```

Inserting Data in Database table using Statement

Inthisprogramwearegoingtoinsertthedatainthedatabasefromourjavaprograminthetablestoredinthedatabase.

Toaccomplishourgoalwefirsthavetomakeaclassnamedas**ServletInsertingData,** whichmust extends theabstract**HttpServlet** class, thenameofthe class should be such that other person can understand what this program is going toperform. The logic of the program will be written inside the **doGet()** method that takestwoarguments,firstis**HttpServletRequest**interfaceandthesecondoneistheHttpServletResponseinterfaceand thismethodcanthrow**ServletException.**

Inside this method call the **getWriter()** method of the **PrintWriter** class. We caninsert thedatainthedatabase only andonly ifthereisaconnectivity between ourdatabase and the java program. To establish the connection between our database and thejava program we first need to call the method **forName(),** which is static in nature of theclass Class. It takes one argument which tells about the database driver we are going touse.Nowusethestaticmethod**getConnection()** ofthe**DriverManager** class.Thismethod takes three arguments

and returns the Connection object. SQL statements areexecutedandresultsarereturnedwithinthecontextofaconnection.Nowyourconnectionhasbeenestablished.Nowusethemethod**createStatement()**oftheConnectionobjectwhichwillreturntheStatementobject.Thisobjectisusedforexecuting a static SQL statement and obtaining the results produced by it. We have toinsert a values into the table so we need to write a query for inserting the values into thetable. This query we will write inside the**executeUpdate()** method of the Statementobject.Thismethodreturnsintvalue.

If the record will get inserted in the table then output will show "record has beeninserted"otherwise"sorry!Failure".

Program

```
Import java.io.*;
Import java.sql.*;
Import javax.servlet.*;
importjavax.servlet.http.*;

public class DataInsertion extends HttpServlet
{
Public void doGet(HttpServletRequestrequest,HttpServletResponseresponse)
Throws ServletException,IOException
{
response.setContentType("text/html");PrintWriterout=response.getWriter();
Stringurl="jdbc:mysql://localhost/zulfiqar?user=root&password=admin";Connectionconn;	ResultSetrs;
try
{
Class.forName("org.gjt.mm.mysql.Driver");conn =
DriverManager.getConnection(url);Statementstatement=conn.createStatement();
Stringquery="insertintoemp_salvalues('Rajesh',15000)";
int i = statement.executeUpdate(query);if(i!=0)
{	out.println("Therecord hasbeeninserted");
}
else
{	out.println("Sorry!Failure");
}
rs=statement.executeQuery("select*fromemp_sal");while(rs.next())
{
out.println("<p><table>"+rs.getString(1)+""+rs.getInt(2)+"</p></table>");
}
```

```
rs.close();statement.close();
}catch(Exceptione)
{
System.out.println(e);
}}}
```

TableinthedatabasebeforeInsertion:

mysql> select*fromemp_sal;Emptyset(0.02sec)

Theoutputof theprogramisgivenbelow:

Rajesh

TableinthedatabaseafterInsertion:

```
mysql>select*fromemp_sal;
+.........+.........-+
|EmpName|salary|
+.........+.........-+
|Rajesh|15000|
+.........+.........-+
1rowinset(0.02sec)
```

Assignment-09

Cookeis

The cookies are small, often encrypted text files, located in browser directories. They are used by web developers to help users navigate their websites efficiently and perform certain functions. Due to their core role of enhancing/enabling usability or site processes, disabling cookies may prevent users from using certain websites. -

Cookies are created when a user's browser loads a particular website. The website sends information to the browser which then creates a text file. Every time the user goes back to the same website, the browser retrieves and sends this file to the website's server.

CookieClass

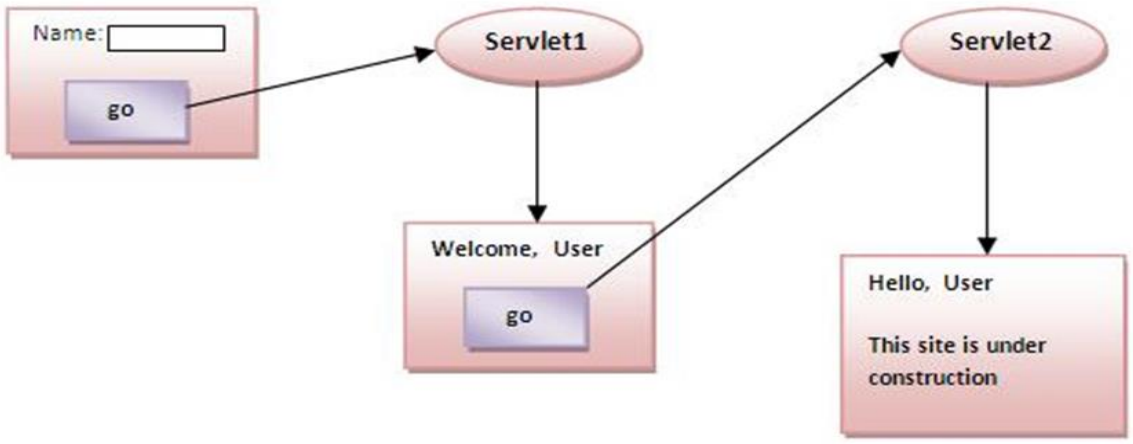
Cookie is the object of the class **javax.servlet.http.Cookie**.
This class is used to create a cookie, it is a small amount of information sent by a servlet to a Web browser, saved by the browser, and later sent back to the server. A cookie's value can uniquely identify a client, so cookies are commonly used for session management.

Cookie objects have the following methods.

Method	**Description**
getMaxAge()	Returns the maximum specified age of the cookie.
getName()	Returns the name of the cookie.
getPath()	Returns the prefix of all URLs for which this cookie is targeted.
getValue()	Returns the value of the cookie.
setMaxAge(int)	Sets the maximum age of the cookie. The cookie will expire after that.
setPath(String)	This cookie should be presented only with requests beginning with this URL.
setValue(String)	Sets the value of the cookie. Values with various special characters (white space,bracketsandparentheses,theequalssign,comma,doublequote,slashes,question marks, the"at"sign, colon, and semicolon)should be avoided.

Simple example on Servlet Cookies

In this example, we are storing the name of the user in the cookie object and accessing it in another servlet. As we know well that session corresponds to the particular user. So if you access it from too many browsers with different values, you will get the different value.



index.html

```
<form action="servlet1" method="post">
Name:<input type="text" name="userName"/>
<br/>
<input type="submit" value="go"/>
</form>
```

FirstServlet.java

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
public class FirstServlet extends HttpServlet {
   public void doPost(HttpServletRequest request, HttpServletResponse response){
   try{
     response.setContentType("text/html");
     PrintWriter out = response.getWriter();
              String n=request.getParameter("userName");
     out.print("Welcome "+n);
     Cookie ck=new Cookie("uname",n);//creating cookie object
     response.addCookie(ck);//adding cookie in the response
     //creating submit button
     out.print("<form action='servlet2'>");
```

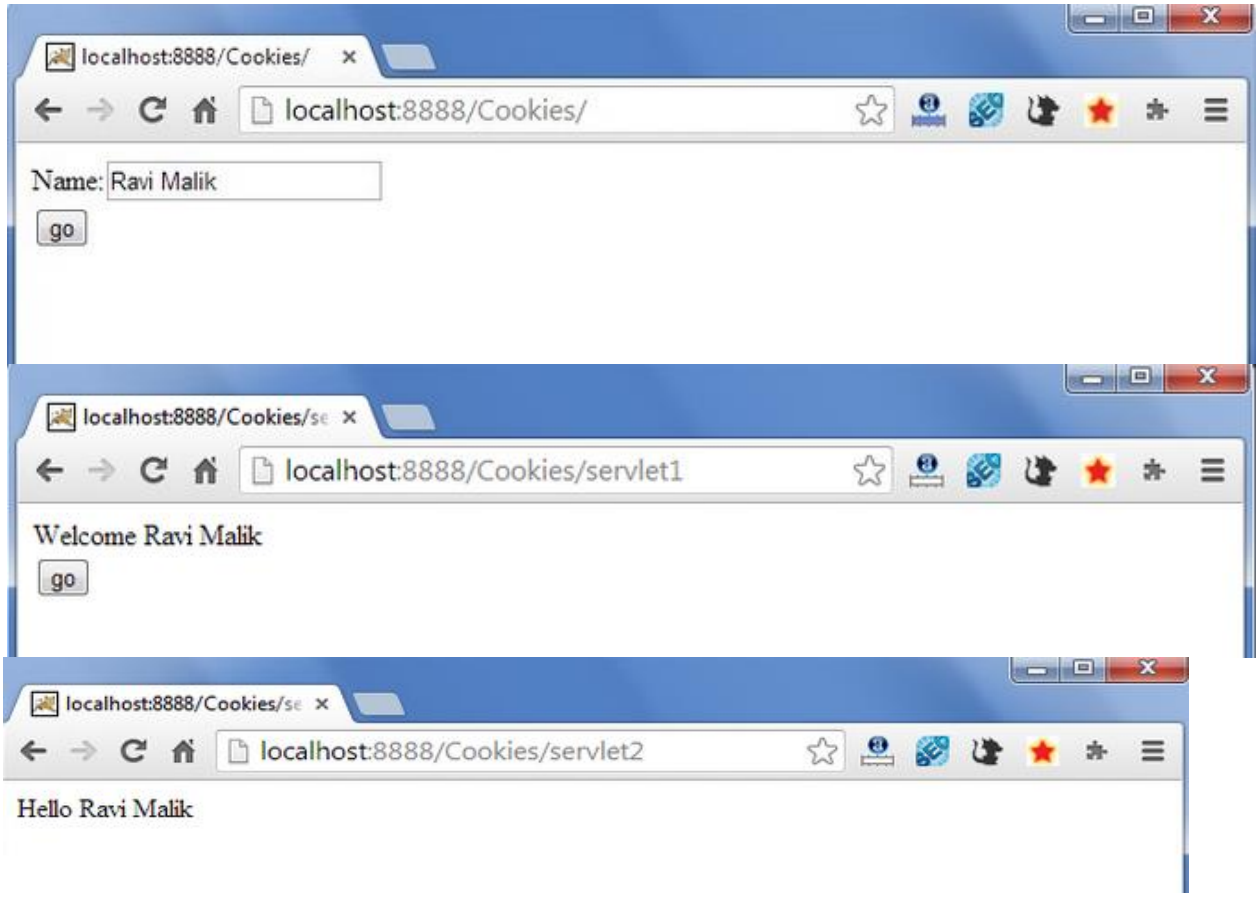
```
                out.print("<input type='submit' value='go'> </form>");
        out.close();
      }catch(Exception e){System.out.println(e);}
  }
}
```

SecondServlet.java

```
import java.io.*;  import javax.servlet.*;  import javax.servlet.http.*;
public class SecondServlet extends HttpServlet {
  public void doPost(HttpServletRequest request, HttpServletResponse response){
    try{  response.setContentType("text/html");
      PrintWriter out = response.getWriter();
                Cookie ck[]=request.getCookies();
      out.print("Hello "+ck[0].getValue());
      out.print("This site is under Construction");
      out.close();
     }catch(Exception e){System.out.println(e);}
  }
}
```

web.xml

```
<web-app>
<servlet>
<servlet-name>s1</servlet-name>
<servlet-class>FirstServlet</servlet-class>
<servlet-name>s2</servlet-name>
<servlet-class>SecondServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>s1</servlet-name>
<url-pattern>/servlet1</url-pattern>
<servlet-name>s2</servlet-name>
<url-pattern>/servlet2</url-pattern>
</servlet-mapping>
</web-app>
```

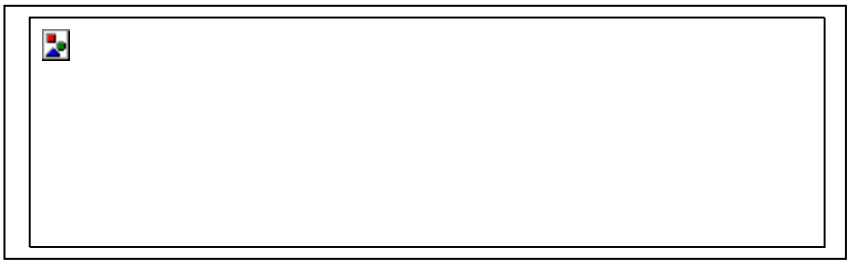


Program:/*Display path of the cookie*/

```
import javax.servlet.*;
import javax.servlet.http.*;
public class ReadCookiesextends HttpServlet
{       public void doGet(HttpServletRequest request,HttpServletResponse response) throws ServletException,
java.io.IOException
        {   Cookie cookie=null;
            Boolean newCookie=false;
            if (cookie==null)
            {       newCookie=true;
                    cookie=newCookie("Cookies",""+getNextCookieValue());
                    cookie.setPath(request.getContextPath());
                    response.addCookie(cookie);
            }response.setContentType("text/html");
            java.io.PrintWriter out=response.getWriter();
            out.println("<html>");out.println("<head>");
            out.println("<title>Read Cookie</title>");
            out.println("</head>");out.println("<body>");
            out.println("<h2> Our Cookie named \"Cookies\"information</h2>");
```

```
        if (newCookie)
        {    out.println("CookiePath:" +cookie.getPath()+"<br>");
        }
        out.println("</body> </html>");
        out.close();
    }
    private long getNextCookieValue()
    {        return new java.util.Date().getTime();
    }
    Public void doPost(HttpServletRequest request,HttpServletResponse response)throws ServletException,
    java.io.IOException
    {        doGet(request,response);
    }
}
```

Output:

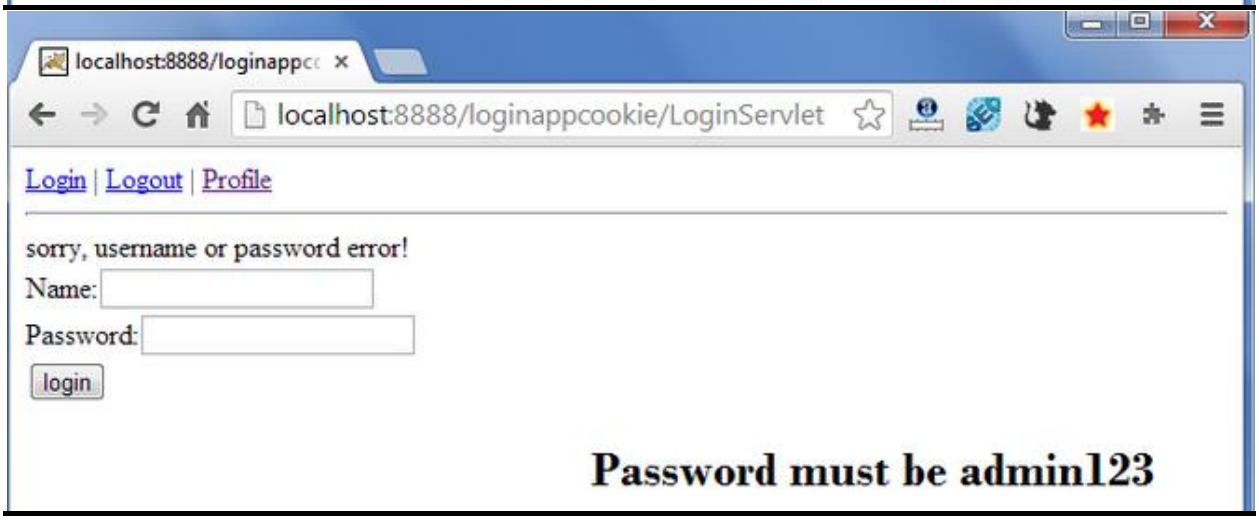


Program1:

- create a login and logout example using servlet cookies.
- In this example, we are creating 3 links: login, logout and profile. User can't go to profile page until he/she is logged in. If user is logged out, he need to login again to visit profile.
- In this application, we have created following files.
    1) index.html
    2) link.html
    3) login.html
    4) LoginServlet.java
    5) LogoutServlet.java
    6) ProfileServlet.java
    7) web.xml


Servlet Login Example
localhost:8888/loginappcookie/
Welcome to Login App by Cookie
Login| Logout| Profile


localhost:8888/loginappcc
localhost:8888/loginappcookie/ProfileServlet
Login | Logout | Profile
Please login first
Name:
Password:
login
You can't visit profile page directly

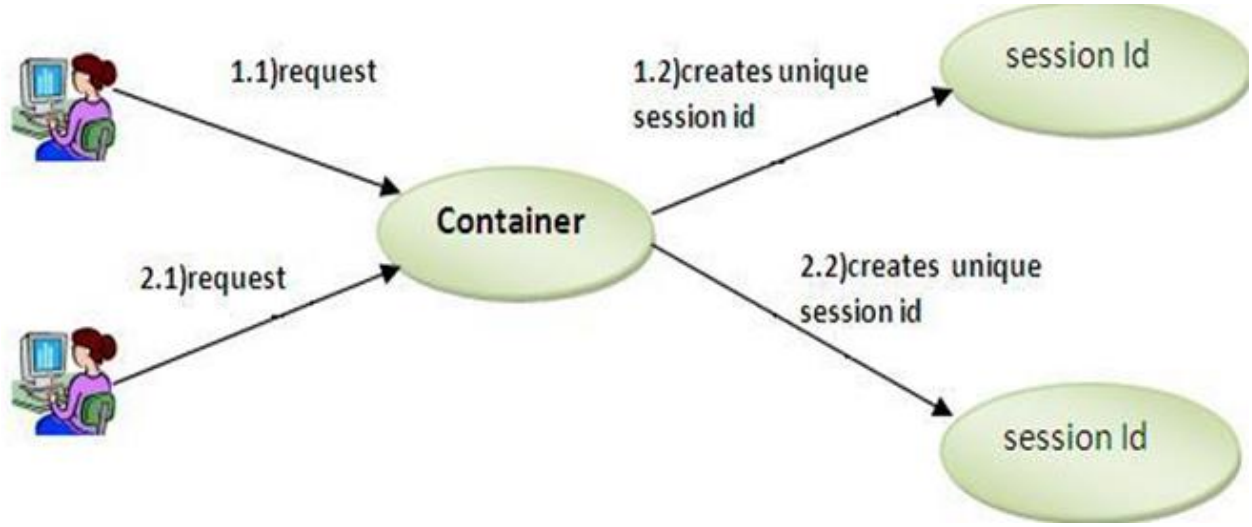

localhost:8888/loginappcc
localhost:8888/loginappcookie/LoginServlet
Login | Logout | Profile
sorry, username or password error!
Name:
Password:
login
Password must be admin123

Experiment – 10
SessionTracking

HttpSession interface

Container creates a session id for each user. The container uses this id to identify the particular user. An object of HttpSession can be used to perform two tasks:

1. bind objects
2. view and manipulate information about a session, such as the session identifier, creation time, and last accessed time.



How to get the HttpSession object ?

The HttpServletRequest interface provides two methods to get the object of HttpSession:

1. public HttpSession getSession():Returns the current session associated with this request, or if the request does not have a session, creates one.
2. public HttpSession getSession(boolean create):Returns the current HttpSession associated with this request or, if there is no current session and create is true, returns a new session. If create is false returns existing session otherwise null.

index.html

```
<html><body>
<form action="servlet1">
Name:<input type="text" name="userName"/><br/>
<input type="submit" value="go"/>
</form> </body></html>
```

FirstServlet.java

```
import java.io.*;  import javax.servlet.*;  import javax.servlet.http.*;
public class FirstServlet extends HttpServlet {
  public void doGet(HttpServletRequest request, HttpServletResponse response){
    try{
                response.setContentType("text/html");
```

```
PrintWriter out = response.getWriter();
String n=request.getParameter("userName");
out.print("Welcome "+n);
HttpSession session=request.getSession();
session.setAttribute("uname",n);
out.print("<a href='servlet2'>visit</a>");
out.close();
    }catch(Exception e){System.out.println(e);}
  }
}
```

SecondServlet.java

```
import java.io.*;  import javax.servlet.*;  import javax.servlet.http.*;
public class SecondServlet extends HttpServlet {
   public void doGet(HttpServletRequest request, HttpServletResponse response)
    try{
                    response.setContentType("text/html");
PrintWriter out = response.getWriter();
//Getting the value from the hidden field
String n=request.getParameter("uname");
out.print("Hello "+n);
out.close();
    }catch(Exception e){System.out.println(e);}
  }
}
```

web.xml

```
<web-app>
<servlet>
<servlet-name>s1</servlet-name>
<servlet-class>FirstServlet</servlet-class>
<servlet-name>s2</servlet-name>
<servlet-class>SecondServlet</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>s1</servlet-name>
<url-pattern>/servlet1</url-pattern>
<servlet-name>s2</servlet-name>
<url-pattern>/servlet2</url-pattern>
</servlet-mapping></web-app>
```

Q1. Demonstrate session life cycle in JSP with cookies.
Q2. WAP to demonstrate the methods for managing the state in the HttpSession.
Q3. WAP to design simple shopping cart using sessions.
Q4. Write a JSP program to display the value present in a textfield.

Scriptlet tag that prints the user name

Index.html

```
<html>
<body>
<formaction="welcome.jsp">
<inputtype="text"name="uname">
<inputtype="submit"value="go"><br/>
</form>
</body>
</html>
```

welcome.jsp

```
<html>
<body>
<%
Stringname=request.getParameter("uname");out.print("welcome"+name);
%>
</form>
</body>
</html>
```

JSP expression tag that prints the username

Index.jsp

```
<html>
<body>
<formaction="welcome.jsp">
<inputtype="text"name="uname"><br/>
<inputtype="submit"value="go">
</form>
</body>
</html>
```

welcome.jsp

```
<html>
```

```
<body>
<%="Welcome"+request.getParameter("uname")%>
</body>
</html>
```

JSP declaration tag that declares method

Index.jsp

```
<html><body>
<%!
intcube(int n){returnn*n*n*;
}
%>
<%="Cubeof3is:"+cube(3)%>
</body>
</html>
```

index.html

JSPrequestimplicitobject

```
<formaction="welcome.jsp">
<inputtype="text"name="uname">
<inputtype="submit"value="go"><br/>
</form>
```

welcome.jsp

```
<%
Stringname=request.getParameter("uname");
out.print("welcome"+name);
%>
```

JSPresponseimplicitobject

index.html

```
<formaction="welcome.jsp">
<inputtype="text"name="uname">
<inputtype="submit"value="go"><br/>
</form>
```

welcome.jsp

```
<%      response.sendRedirect("http://www.google.com");
%>
```

JSP config implicit object

index.html

```
<formaction="welcome">
<inputtype="text"name="uname">
<inputtype="submit"value="go"><br/>
</form>
```

web.xml

```
<web-app>
<servlet>
<servlet-name>sonoojaiswal</servlet-name>
<jsp-file>/welcome.jsp</jsp-file>
<init-param>
<param-name>dname</param-name>
<param-value>sun.jdbc.odbc.JdbcOdbcDriver</param-value>
</init-param>
</servlet>
<servlet-mapping>
<servlet-name>sonoojaiswal</servlet-name>
<url-pattern>/welcome</url-pattern>
</servlet-mapping>
</web-app>
```

welcome.jsp

```
<%
out.print("Welcome"+request.getParameter("uname"));

Stringdriver=config.getInitParameter("dname");out.print("driver nameis="+driver);
%>
```

JSPapplicationimplicitobject

index.html

```
<formaction="welcome">
<inputtype="text"name="uname">
<inputtype="submit"value="go"><br/>
</form>
```

web.xml

```
<web-app>

<servlet>
```

```
<servlet-name>sonoojaiswal</servlet-name>
<jsp-file>/welcome.jsp</jsp-file>
</servlet>

<servlet-mapping>
<servlet-name>sonoojaiswal</servlet-name>
<url-pattern>/welcome</url-pattern>
</servlet-mapping>

<context-param>
<param-name>dname</param-name>
<param-value>sun.jdbc.odbc.JdbcOdbcDriver</param-value>
</context-param>

</web-app>
```

welcome.jsp

```
<%

out.print("Welcome"+request.getParameter("uname"));

Stringdriver=application.getInitParameter("dname");out.print("driver nameis="+driver);

%>
```

sessionimplicitobject

index.html

```
<html><body>
<formaction="welcome.jsp">
<inputtype="text"name="uname">
<inputtype="submit"value="go"><br/>
</form>
</body></html>
```

welcome.jsp

```
<html><body>
<%
Stringname=request.getParameter("uname");out.print("Welcome "+name);session.setAttribute("user",name);
<ahref="second.jsp">secondjsppage</a>
%>
```

```
</body>          </html>
```

second.jsp

```
<html><body>
<%
String name=(String)session.getAttribute("user");out.print("Hello"+name);

%>
</body>          </html>
```

pageContextimplicitobject

index.html

```
<html><body>
<formaction="welcome.jsp">
<inputtype="text"name="uname">
<inputtype="submit"value="go"><br/>
</form>
</body>          </html>
```

welcome.jsp

```
<html><body>
<%
Stringname=request.getParameter("uname");out.print("Welcome"+name);
pageContext.setAttribute("user",name,PageContext.SESSION_SCOPE);

<ahref="second.jsp">secondjsppage</a>
%>
</body>          </html>
```

second.jsp

```
<html>
<body>
<%
Stringname=(String)pageContext.getAttribute("user",PageContext.SESSION_SCOPE);out.print("Hello"+name)
;%>
</body>
</html>
```

Exception Handling in JSP

index.jsp

```
<formaction="process.jsp">
No1:<inputtype="text"name="n1"/><br/><br/>No1:<inputtype="text"name="n2"/><br/><br/>
<inputtype="submit"value="divide"/>
</form>
```

process.jsp

```
<%@pageerrorPage="error.jsp" %>
<%
Stringnum1=request.getParameter("n1");Stringnum2=request.getParameter("n2");

int a=Integer.parseInt(num1);int b=Integer.parseInt(num2);intc=a/b;
out.print("divisionofnumbersis:"+c);

%>error.jsp
<%@pageisErrorPage="true" %>
<h3>Sorryanexceptionoccured!</h3>Exceptionis:<%= exception%>
```

JspIncludeDirective

```
<html>
<body>
<%@includefile="header.html"%>

Todayis:<%=java.util.Calendar.getInstance().getTime() %>
</body>
</html>
```

JSP Taglib directive

```
<html>
<body>
<%@tagliburi="http://www.javatpoint.com/tags"prefix="mytag"%>
<mytag:currentDate/>
</body></html>
```

jsp:forward

index.jsp

```
<html>
<body>
<h2>thisisindexpage</h2>
```

```
<jsp:forwardpage="printdate.jsp">
<jsp:paramname="name"value="javatpoint.com"/>
</jsp:forward>
</body>
</html>
```

printdate.jsp

```
<html>
<body>
<%out.print("Todayis:"+java.util.Calendar.getInstance().getTime());%>
<%=request.getParameter("name")%>
</body>
</html>
```

Jsp include directive

index.jsp

```
<h2>thisisindexpage</h2>
<jsp:includepage="printdate.jsp"/>
<h2>endsectionofindexpage</h2>
```

printdate.jsp

```
<%      out.print("Todayis:"+java.util.Calendar.getInstance().getTime());
%>
```

<u>AdditionalListofPrograms:</u>

1. Write a JSP program to display the value present in a text field.
2. WAP to display cookie value, cookie age and cookie path.
3. WAP in JSP file to set and then display the cookie.
4. WAP to create and store value of cookie.

MINI Project Using Servlet/JSP and Sql

Problem Statement will be provided by the respective instructors.