

WEEK2:

AIM: Creating the cart page and registration pages.

<!--Program for creating a CART page contains the details about the books which are added to the cart. The cart page contains the fields: Book, name, Price, Quantity and Amount

cart.html

```
<html>
```

```
<body >
```

```
<table cellspacing="50" cellpadding="20">
```

```
<tr>
```

```
<td>Book name</td>
```

```
<td>price</td>
```

```
<td>quantity</td>
```

```
<td>amount</td>
```

```
</tr>
```

```
<tr>
```

```
<td>java2</td>
```

```
<td>$35.5</td>
```

```
<td>2</td>
```

```
<td>$70</td>
```

```
</tr>
```

```
<tr>
```

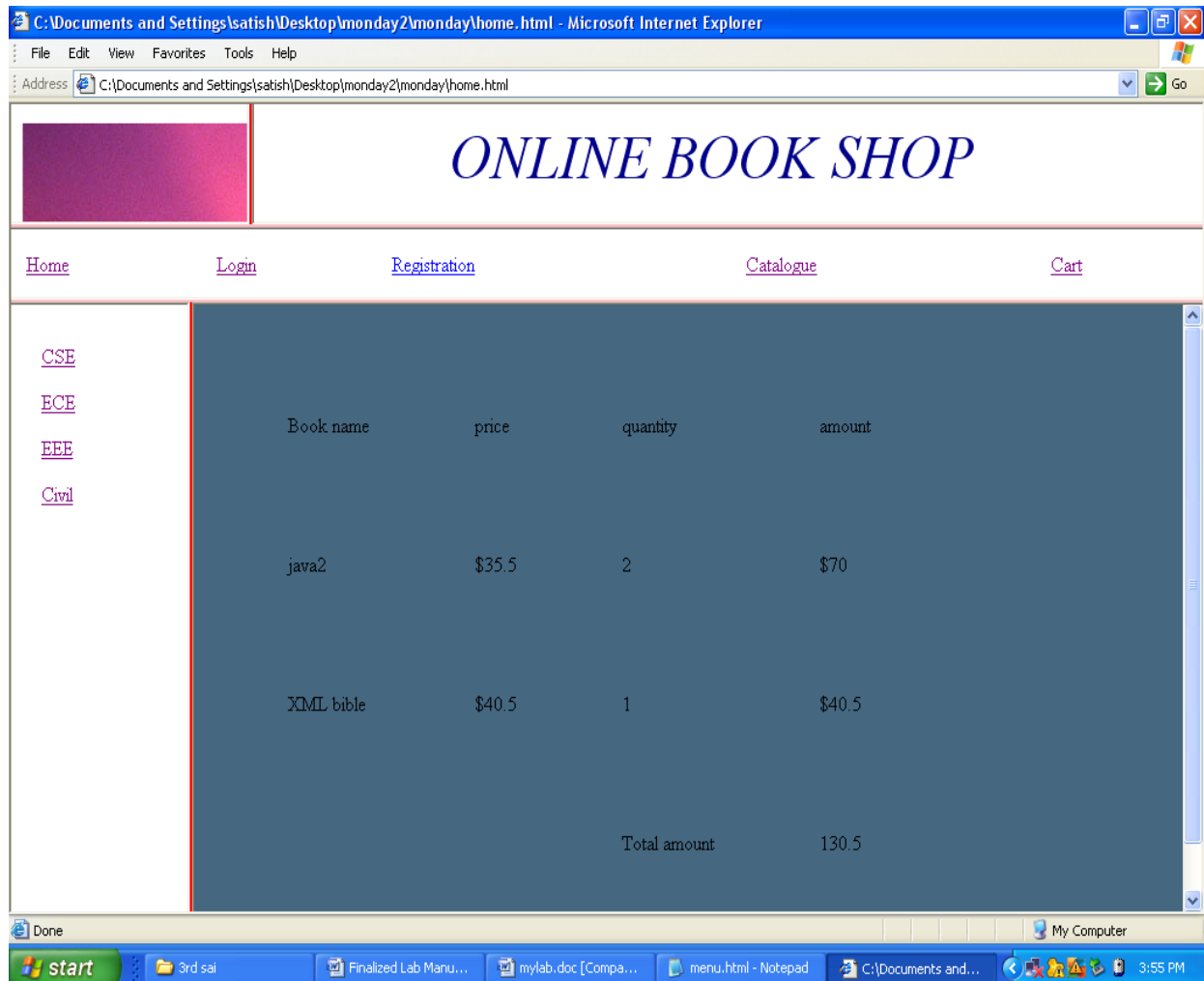
```
<td>XML bible</td>
```

```
<td>$40.5</td>
```

```
<td>1</td>
```

```
<td>$40.5</td>
</tr> <tr>
<td></td>
<td></td>
<td>Total amount </td>
<td>130.5 </td>
</tr>
</body>
</html>
```

Output:



WEEK 3:**AIM: VALIDATION:**

Write JavaScript to validate the following fields of the above registration page.

1. Name (Name should contains alphabets and the length should not be less than 6 characters).
2. Password (Password should not be less than 6 characters length).
3. E-mail id (should not contain any invalid and must follow the standard pattern name@domain.com)
4. Phone number (Phone number should contain 10 digits only).

Note: You can also validate the login page with these parameters.

<!--Program for validating the username (Name should contains alphabets and the length should not be less than 6 characters).

1) NAME VALIDATION

```
<html>

<head>

<title>validating a textbox</title>

<script language="javascript">

function button_click()

{

var fname=f.fname.value;

var spchar = "!@#$$%^&*()+=-[]\\';,./{}|\"':<>?~";

var number="0123456789";

if(fname.length<6)

{
```

```

alert("first name atleast 6 characters");

}

if(fname!="")

{

for (var i=0;i<fname.length;i++)

{

if(spchar.indexOf(fname.charAt(i)) != -1)

{

alert ("Userid should not have special characters");

f.fname.value=" ";

return false;

}

}

}

if(fname!="")

{

for (var i=0;i<fname.length;i++)

{

if(number.indexOf(fname.charAt(i)) != -1)

{

alert ("Userid should not have numbers");

f.fname.value="";

return false;

}

}

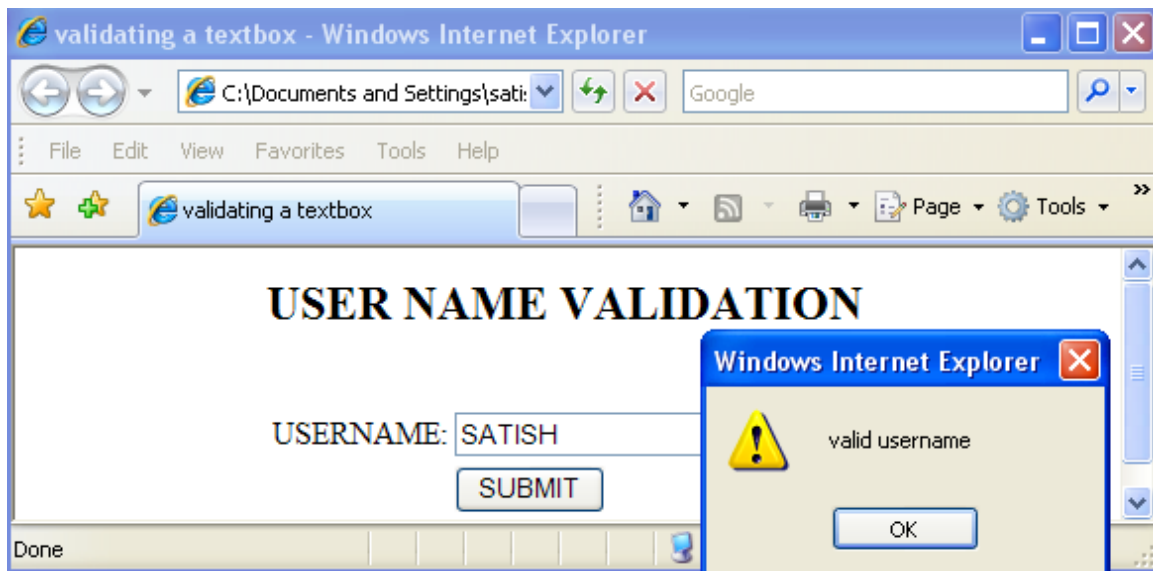
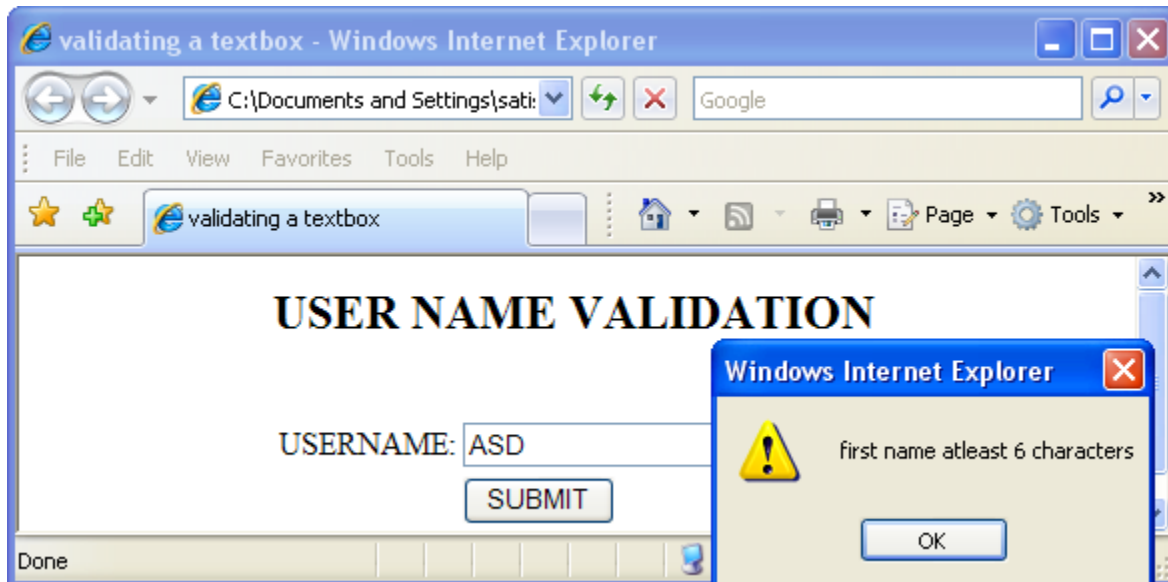
}

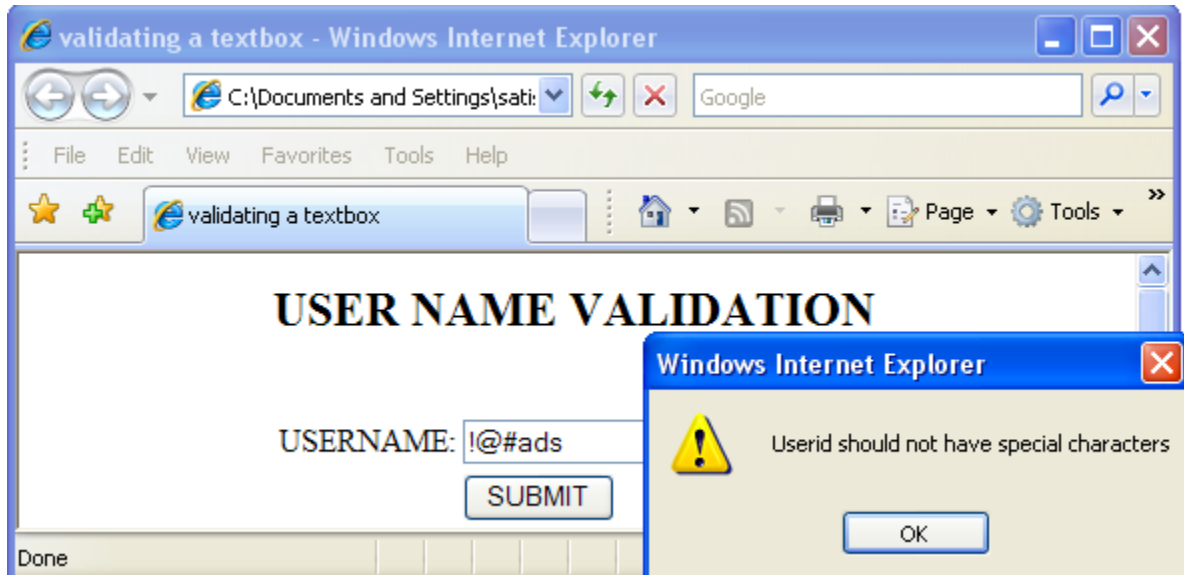
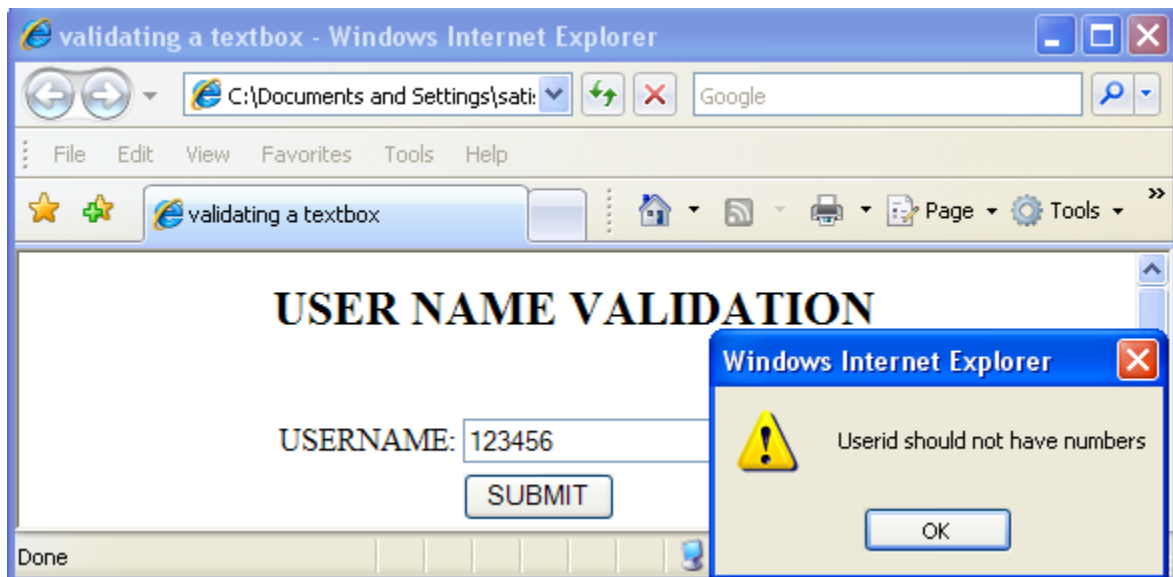
if(fname.length>=6)

```

```
alert("valid username");
}
}
</script>
<body>
<center>
<form name="f">
<table>
<h2 align="center">USER NAME VALIDATION</h2>
<tr>
<td align="right">USERNAME:</td>
<td><input type="text" name="fname" maxlength=20 size=30></td>
</tr>
<br>
<tr>
<td></td>
<td><input type="button" value="SUBMIT" onClick="button_click()"></td></tr>
</form>
</table>
</center>
</body>
</html>
```

OUTPUT:





2) <!--Program for password field validation(should not be less than 6 characters)

(pwdvalid.html)-->

```
<html>

<head>

<title>PASSWORD VALIDATION</title>

<script language="javascript">

function pass()

{

var pw=f.pw.value;

var cpw=f.cpw.value;

if(pw.length<6)

{

alert("PASSWORD MUST BE 6 CHARACTERS");

}

if(pw!=cpw)

{

alert("PASSWORD DON'T MATCH");

}

else if(pw.length>=6)

alert("PASSWORD VALIDATION SUCCESS");

}

</script>

</head>
```

```

<body>

<form name="f">

<table align="center">

<h2 align="center">PASSWORD VALIDATION</h2>

<br>

<tr>

<td align="right">PassWord:</td>

<td align="left"><input type="PassWord" maxlength="10" size="30" name="pw"></td>

</tr>

<tr>

<td align="right">Confirm PassWord:</td>

<td align="left"><input type="PassWord" maxlength="10" size="30" name="cpw"></td>

</tr>

<tr>

<td></td>

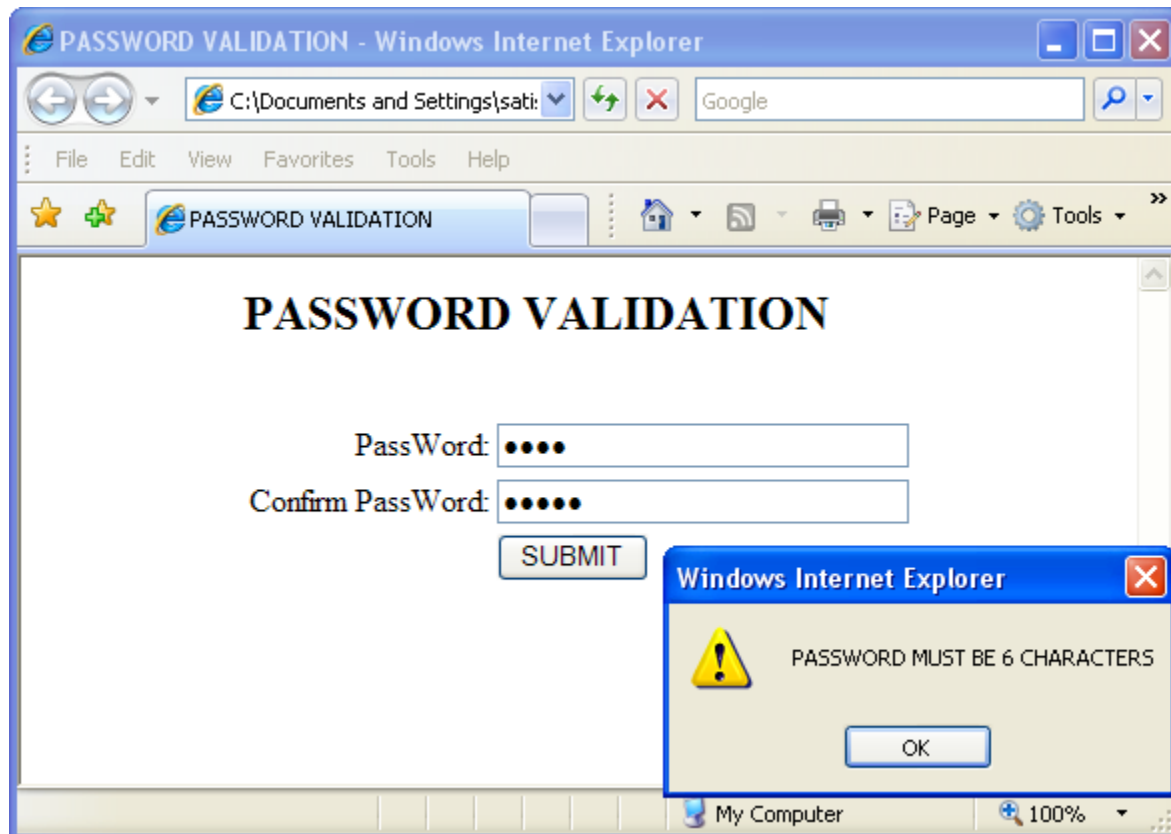
<td><input type="button" value="SUBMIT" onClick="pass()"/></td>

</tr>

</table> </form> </body> </html>

```

Output:



3) <!--Program for E-mail id validation(should not contain any invalid and must follow the standard pattern-)(emailvalid.html)-->

```

<html>

<head>

<title>EMAIL ID VALIDATION</title>

<script language="javascript">

function check()

{

if(f.txtname.value.indexOf('@')== -1)

{

alert("invalid");

}

else

if((f.txtname.value.indexOf('@')==0)|| (f.txtname.value.indexOf('.')==0)|| (f.txtname.value.indexOf('@'))>

(f.txtname.value.indexOf('.')))

{

alert("invalid email id");

f.txtname.value="";

}

else

alert("valid email id");

}

</script>

</head>

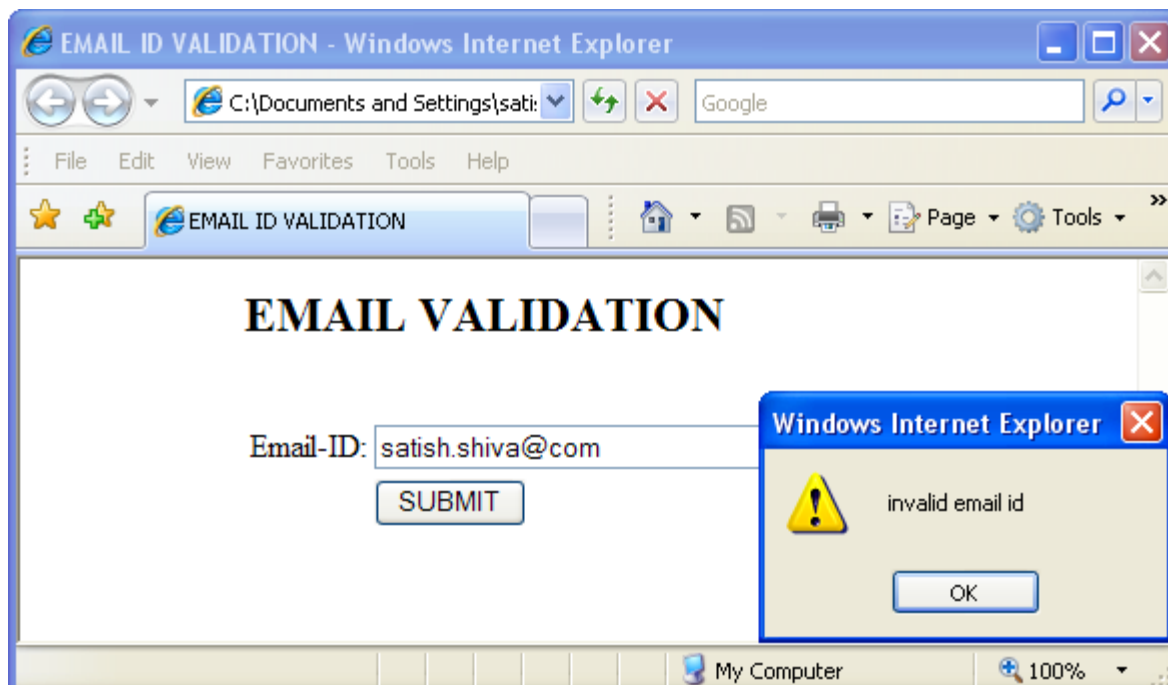
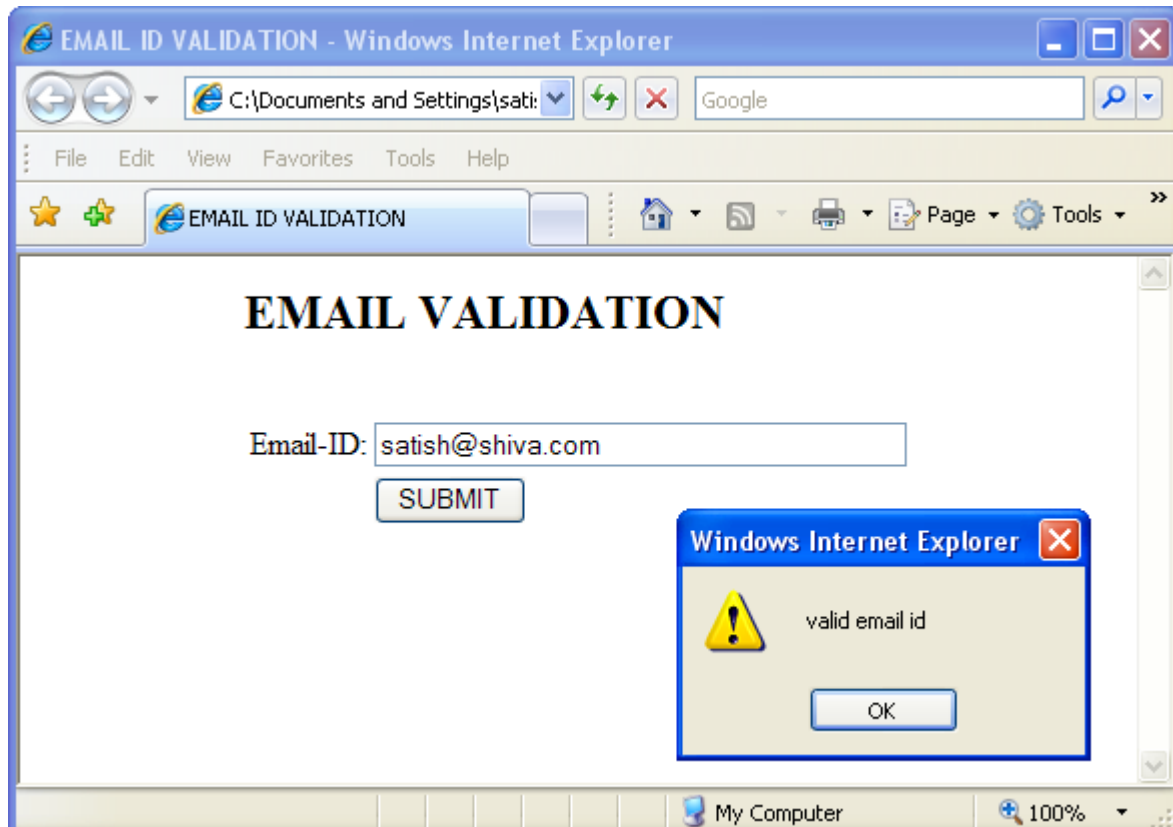
<body>

<form name="f">

```

```
<table align="center">
<h2 align="center">EMAIL VALIDATION</h2>
<br>
<tr>
<td align="right">Email-ID:</td>
<td align="left"><input type="text" name="txtname" size=40>
</tr>
<tr>
<td></td>
<td><input type="button" value="SUBMIT" onClick="check()"/></td>
</tr>
</table>
</form>
</body>
</html>
```

Output:



4) <!--Program for Phone number validation(Phone number should contain 10 digits only)(phnvalid.html)-->

```

<html>

<head>

<title>PHONE NUMBER VALIDATION</title>

<script language="javascript">

function phno()

{

var ph=f.ph.value;

if(ph.length==10)

{

if(isNaN(ph))

alert("It is not a valid Phone number");

else

alert("It is a valid Phone number");

}

else

alert("Please enter a valid phone number");

}

</script>

</head>

<body>

<form name="f">

<table align="center">

<br><br><br><br>

<h2 align="center">PHONE NUMBER VALIDATION</h2>

```

```
<br>
```

```
<tr>
```

```
<td align="right">PHONE NO:</td>
```

```
<td align="left"><input type="text" maxlength="15" size="30" name="ph"></td>
```

```
</tr>
```

```
<tr>
```

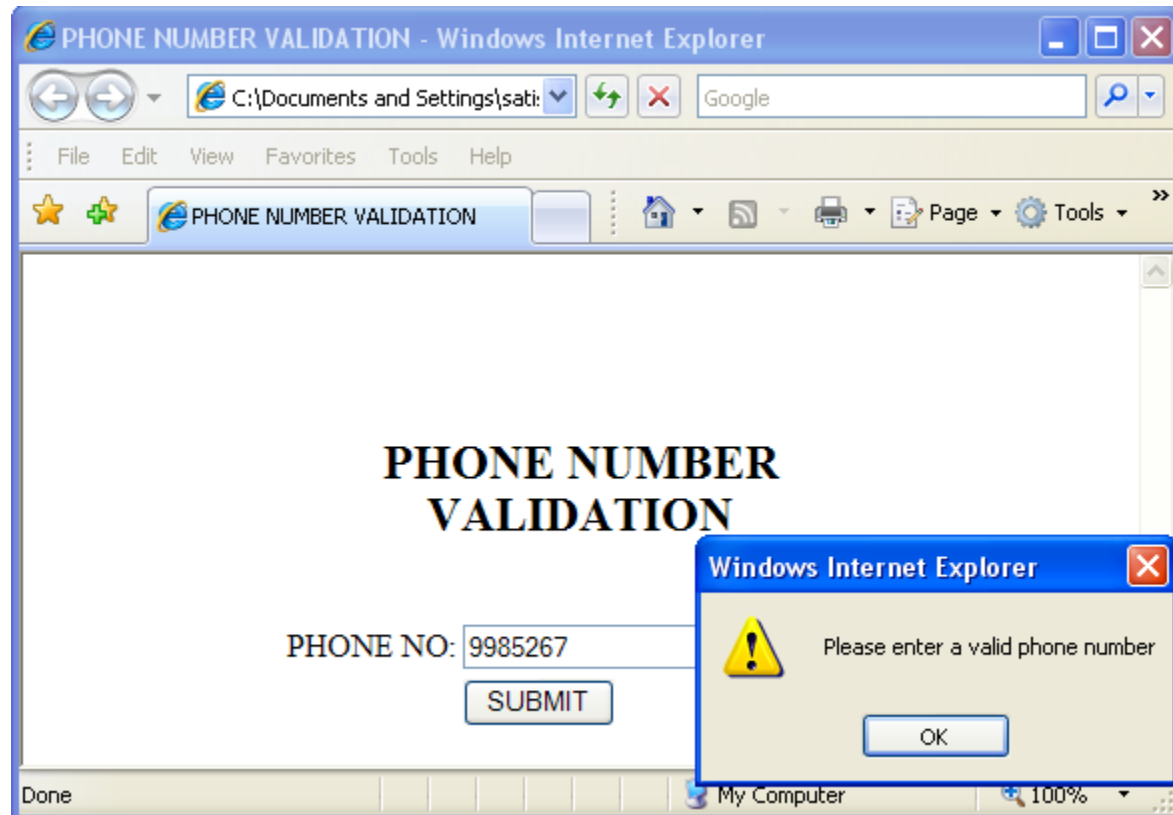
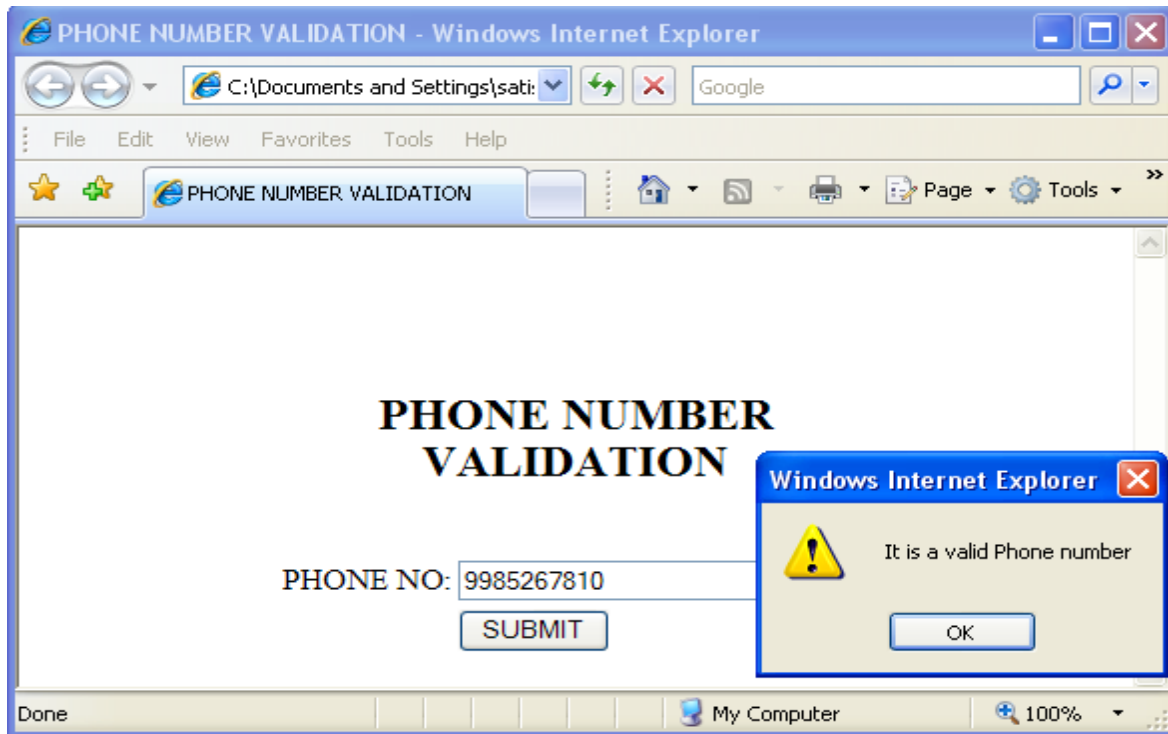
```
<td></td>
```

```
<td><input type="button" value="SUBMIT" onClick="phno()"/></td>
```

```
</tr>
```

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


Output:



WEEK-4:

AIM: Design a web page using **CSS (Cascading Style Sheets)** which includes the following:

1. Use different font, styles:

In the style definition you define how each selector should work (font, color etc.).

Then, in the body of your pages, you refer to these selectors to activate the styles.

<!--Program for creating inline style sheets

1) Inline.html

```
<head>
```

```
<title>inline style sheet</title>
```

```
</head>
```

```
<body>
```

```
normal bold:<b>web technologies</b><br>
```

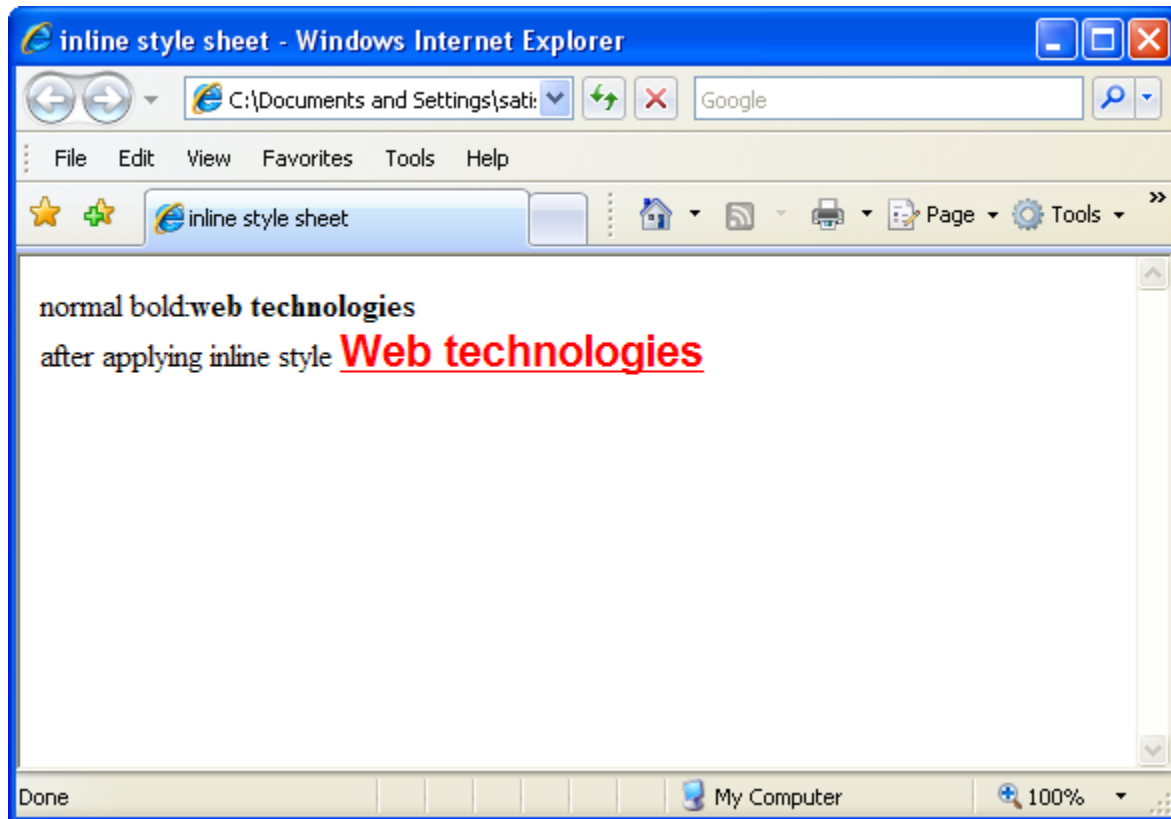
after applying inline style

```
<b style="color:red;font-size:22px;font-family:arial;text-decoration:underline">Web technologies</b>
```

```
</body>
```

```
</html>
```

Output:



2. Set a background image for both the page and single elements on the page.

<!--Program for creating the background image(image.html)-->

```
<html>

<head>

<style type="text/css">

body

{

background-image: url("C:\Documents and Settings\All Users\Documents\My Pictures\Sample
Pictures\Sunset.jpg");

background-repeat:no-repeat

}

</style>

</head>

<body>

<center>

<font color=fuchsia>

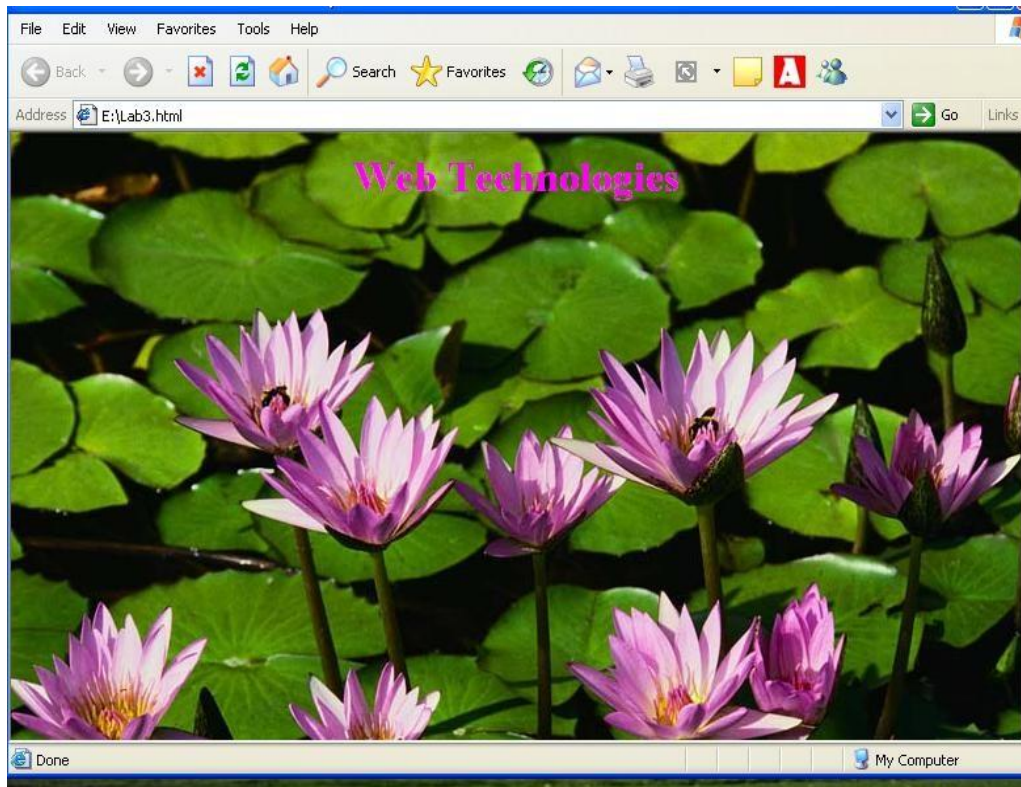
<h1>WebTechnologies</h1>

</center>

</body>

</html>
```

Output:



3. Control the repetition of the image with the background-repeat property.

As background-repeat: repeat

Tiles the image until the entire page is filled, just like an ordinary background image in plain HTML.

<!--Program for image repetition

```
<html>

<head>

<style type="text/css">

body{

background-image:url("C:\Documents and Settings\All Users\Documents\My Pictures\Sample
Pictures\Winter.jpg");

background-repeat:repeat

}

</style>

</head>

<body>

<center>

<font color=fuchsia>

<h1>satish</h1>

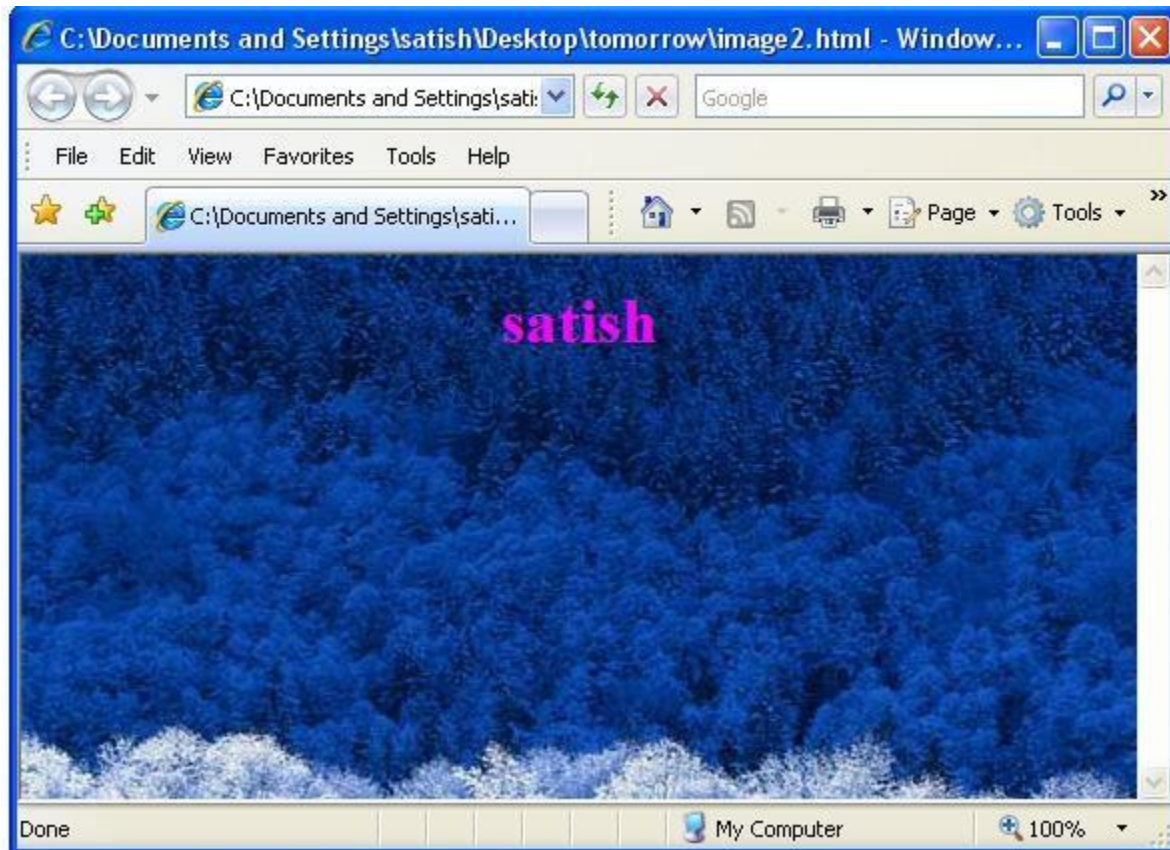
</font>

</center>

</body>

</html>
```

Output:



4. Define styles for links as **A:link**

A:visited

A:active

A:hover

<!--Defining styles using CSS(external.html)-->

<html>

<head>

<title>external style sheet</title>

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

</head>

<body bgcolor="black">

**normal bold : Web technologies
after applied embedded style :**

**<b class="headline">Web Technologies
**

CROSS LINK

**
**

**HELP LINK

**

another LINK

</body>

</html>

style.css

```
b.headline {color:green; font-size:22px; font-family:arial; text-decoration:underline}
```

```
.xlink {cursor:crosshair;color:cyan}
```

```
.hlink{cursor:help;color:purple}
```

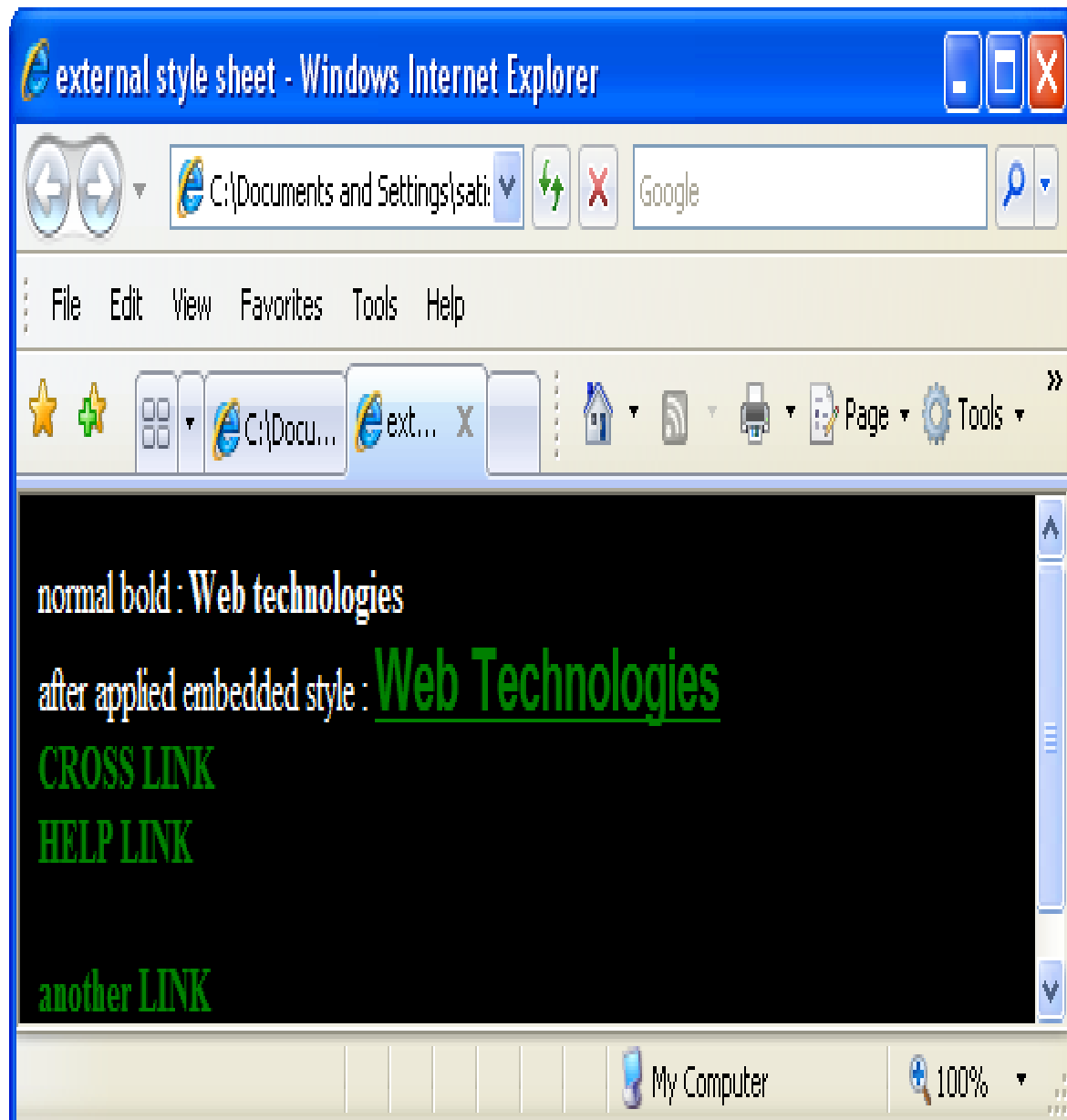
```
A:link {text-decoration:none;color:green} /* unvisited link */
```

```
A:visited {text-decoration: none;color:yellow} /* visited link */
```

```
A:active {text-decoration: none;color:blue} /* selected link */
```

```
A:hover {text-decoration: underline; color: red} /* mouse over link */
```

Output:



5. Work with layers:

<!--Creating layers using CSS(layer.html)>

<html>

<head>

<title>Layers Demo</title>

</head>

<body>

LAYER 1 ON TOP

<div style="position:relative;font-size:50px;z-index:2;">LAYER1</div>

<div style="position:relative;top:-50;left:5;color:red;font-size:80px;z-index:1;">LAYER2</div>

LAYER 2 ON TOP

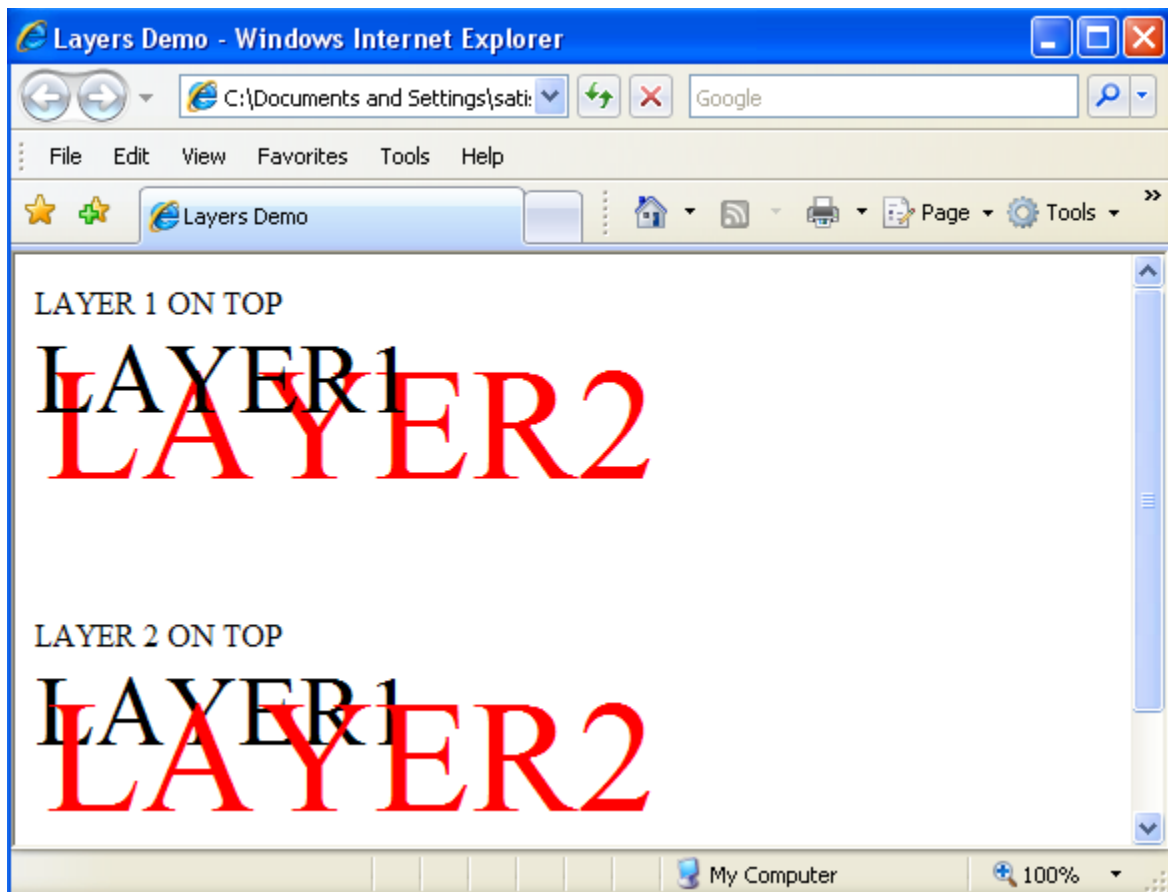
<div style="position:relative;font-size:50px;z-index:3;">LAYER1</div>

```
<div style="position:relative;top:-50;left:5;color:red;font-size:80px;z-index:4;">LAYER2</div>
```

```
</body>
```

```
</html>
```

Output:



6. Add a customized cursor:

<!--Program for representing the various cursors

```
<html>

<head><title>Cursor demo</title>

<style type="text/css">

.link1 { cursor:default}

.link2 { cursor:crosshair}

.link3 { cursor:hand}

.link4 { cursor:move}

.link5 { cursor:text}

.link6 { cursor:wait}

.link7 { cursor:help}

.link8 { cursor:n-resize}

.link9 { cursor:s-resize}

.link10 { cursor:e-resize}

.link11 { cursor:w-resize}

.link12 { cursor:progress}

.link13 { cursor:not-allowed}

.link14 { cursor:no-drop}

.link15 { cursor:all-scroll}

</style></head>

<body>

<b>

<a href="login.html" class="link1">default cursor</a>

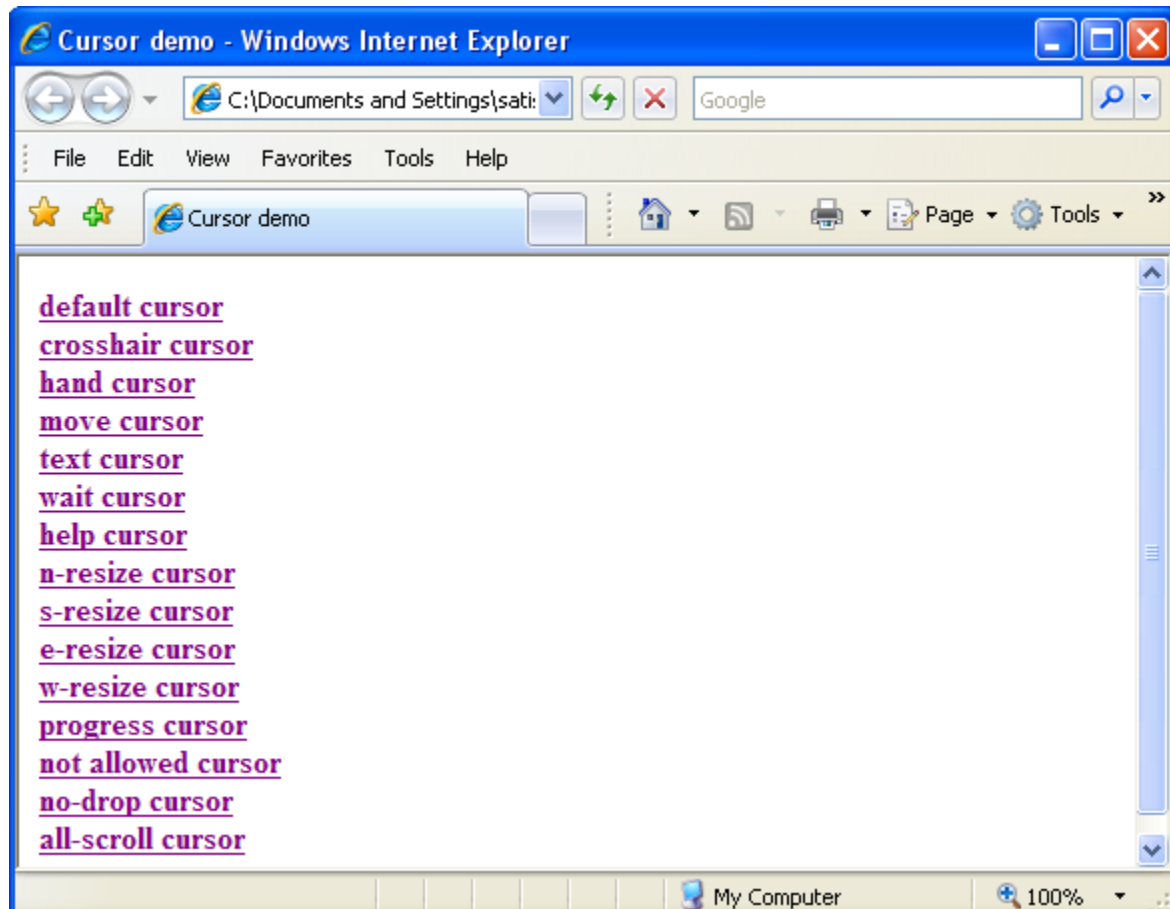
<br>

<a href="login.html" class="link2">crosshair cursor</a>

<br>
```

```
<a href="login.html" class="link3">hand cursor</a> <br>
<a href="login.html" class="link4">move cursor</a> <br>
<a href="login.html" class="link5">text cursor</a>
<br>
<a href="login.html" class="link6">wait cursor</a>
<br>
<a href="login.html" class="link7">help cursor</a>
<br>
<a href="login.html" class="link8">n-resize cursor</a>
<br>
<a href="login.html" class="link9">s-resize cursor</a>
<br>
<a href="login.html" class="link10">e-resize cursor</a>
<br>
<a href="login.html" class="link11">w-resize cursor</a>
<br>
<a href="login.html" class="link12">progress cursor</a>
<br>
<a href="login.html" class="link13">not allowed cursor</a>
<br>
<a href="login.html" class="link14">no-drop cursor</a>
<br>
<a href="login.html" class="link15">all-scroll cursor</a><br></b></body> </html>
```

Output:



WEEK-5:

AIM: Write an XML file which will display the Book information which includes the following:

- 1) Title of the book**
- 2) Author Name**
- 3) ISBN number**
- 4) Publisher name**
- 5) Edition**
- 6) Price**

Write a Document Type Definition (DTD) to validate the above XML file. Display the XML file as follows. The contents should be displayed in a table. The header of the table should be in color GREY. And the Author names column should be displayed in one color and should be capitalized and in bold. Use your own colors for remaining columns. Use XML schemas XSL and CSS for the above purpose.

<!--Program for creating DTD for the XML file(dtd.xml)-->

```
<?xml version="1.0" ?>
```

```
<?xml:stylesheet type="text/xsl" href="Style.xsl"?>
```

```
<!DOCTYPE BOOKS[
```

```
<!ELEMENT BOOKS (INFORMATION)>
```

```
<!ELEMENT INFORMATION (Book)>
```

```
<!ELEMENT Book (book_name,Author_name,ISBN_number,publisher,Edition,Price)>
```

```
<!ELEMENT book_name (#PCDATA)>
```

```
<!ELEMENT Author_name (#PCDATA)>
```

```
<!ELEMENT ISBN_number (#PCDATA)>
```

```
<!ELEMENT publisher (#PCDATA)>
```

```
<!ELEMENT Edition (#PCDATA)>
<!ELEMENT price (#PCDATA)>
]>
<BOOKS>
<INFORMATION>
<Book>
<book_name>XML BIBLE</book_name>
<Author_name>RAJA</Author_name>
<ISBN_number>11530-115</ISBN_number>
<publisher>Pearson</publisher>
<Edition>I-Edition</Edition>
<Price>750</Price>
</Book>
<Book>
<book_name>HTML</book_name>
<Author_name>DIETL</Author_name>
<ISBN_number>11528-115</ISBN_number>
<publisher>wiley</publisher>
<Edition>III-Edition</Edition>
<Price>470</Price>
</Book>
<Book>
<book_name>JAVASCRIPT</book_name>
<Author_name>RANI</Author_name>
<ISBN_number>11525-115</ISBN_number>
<publisher>Techical</publisher>
```

```

<Edition>IV-Edition</Edition>

<Price>270</Price>

</Book>

<Book>

<book_name>WEB TECHNOLOGIES</book_name>

<Author_name>PUNTAMBEKAR</Author_name>

<ISBN_number>11530-115</ISBN_number>

<publisher>Technical</publisher>

<Edition>I-Edition</Edition>

<Price>340</Price>

</Book>

</INFORMATION>

</BOOKS>

```

Style.xsl

```

<?xml version="1.0" ?>

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<xsl:template match="/">

<html>

<body>

<h2 align="center">My BOOK Collection</h2>

<table border="1" align="center">

<tr bgcolor="GRAY">

<th><font color="white"> BOOK_NAME</font></th>

<th><font color="white"> Author</font></th>

<th><font color="white"> ISBN</font></th>

```

```

<th><font color="white"> Publisher</font></th>

<th><font color="white"> Edition</font></th>

<th><font color="white"> Price</font></th>

</tr>

<xsl:for-each select="BOOKS/INFORMATION/Book">

<tr>

<td><xsl:value-of select="book_name"/></td>

<td bgcolor="black"><b><font color="white">

<xsl:value-of select="Author_name"/></font></b></td>

<td><xsl:value-of select="ISBN_number"/></td>

<td><xsl:value-of select="publisher"/></td>

<td><xsl:value-of select="Edition"/></td>

<td><xsl:value-of select="Price"/></td>

</tr>

</xsl:for-each>

</table>

</body>

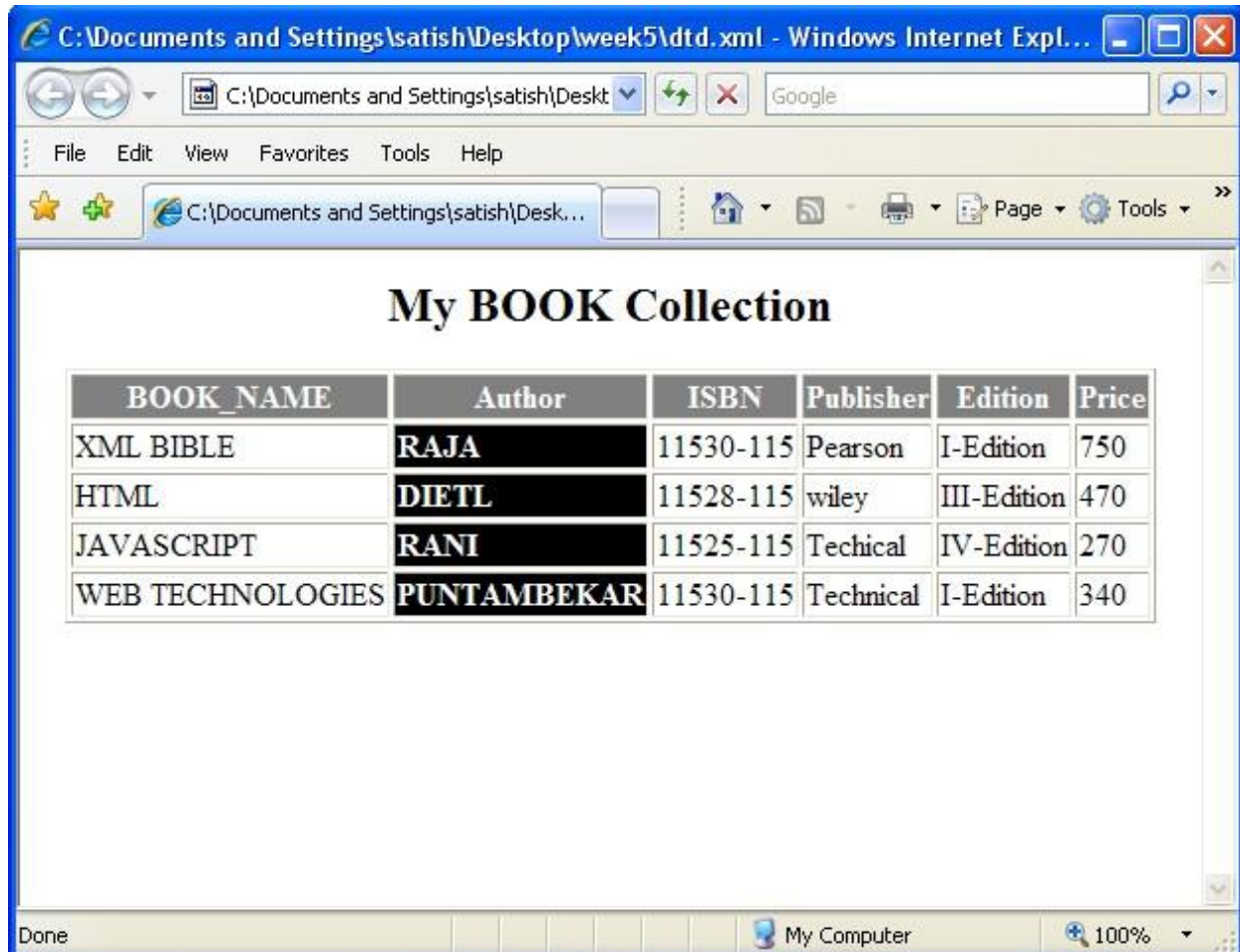
</html>

</xsl:template>

</xsl:stylesheet>

```

Output:



My BOOK Collection

BOOK_NAME	Author	ISBN	Publisher	Edition	Price
XML BIBLE	RAJA	11530-115	Pearson	I-Edition	750
HTML	DIETL	11528-115	wiley	III-Edition	470
JAVASCRIPT	RANI	11525-115	Technical	IV-Edition	270
WEB TECHNOLOGIES	PUNTAMBEKAR	11530-115	Technical	I-Edition	340

Done My Computer 100%

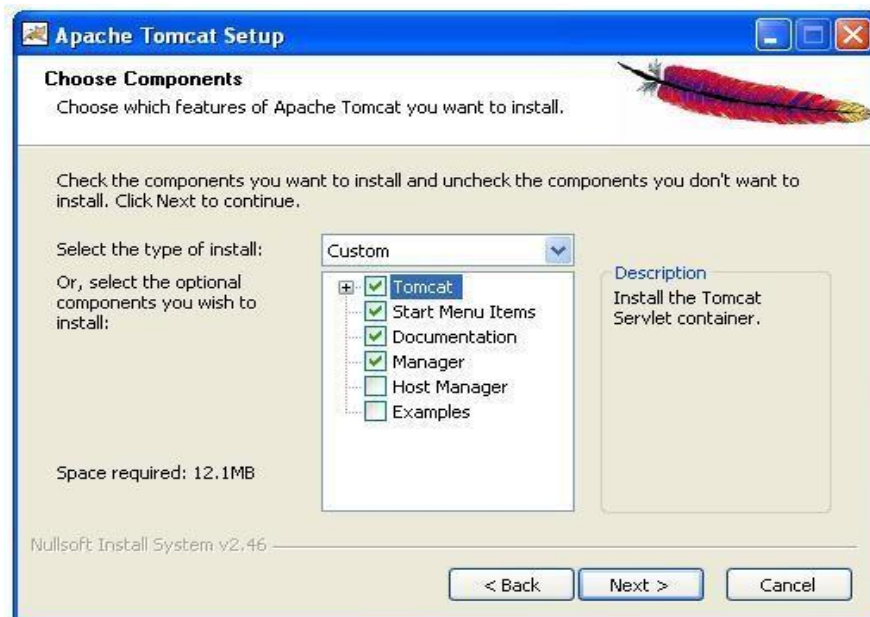
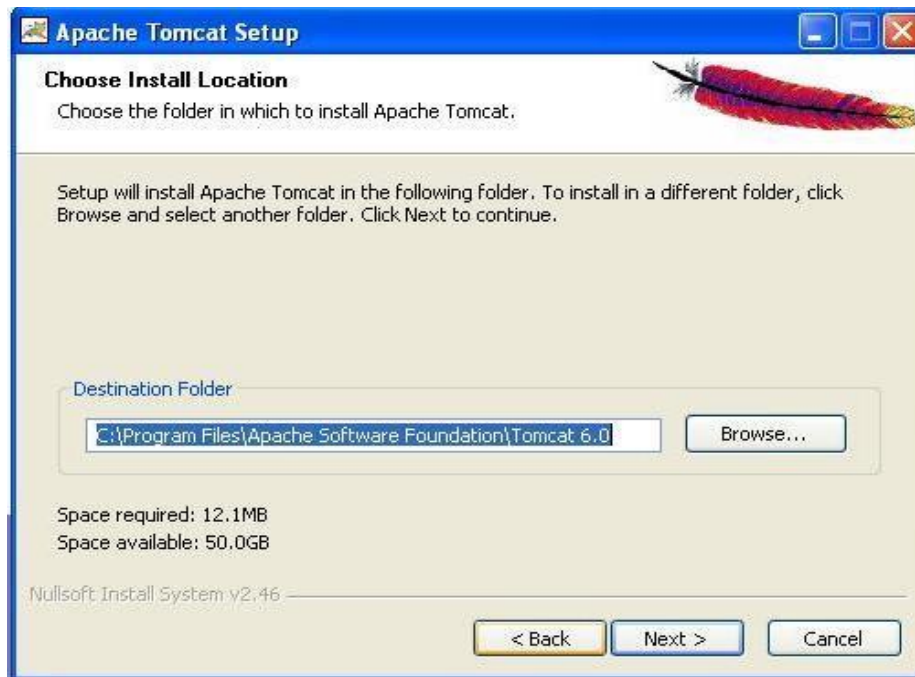
WEEK - 6:**Aim:**

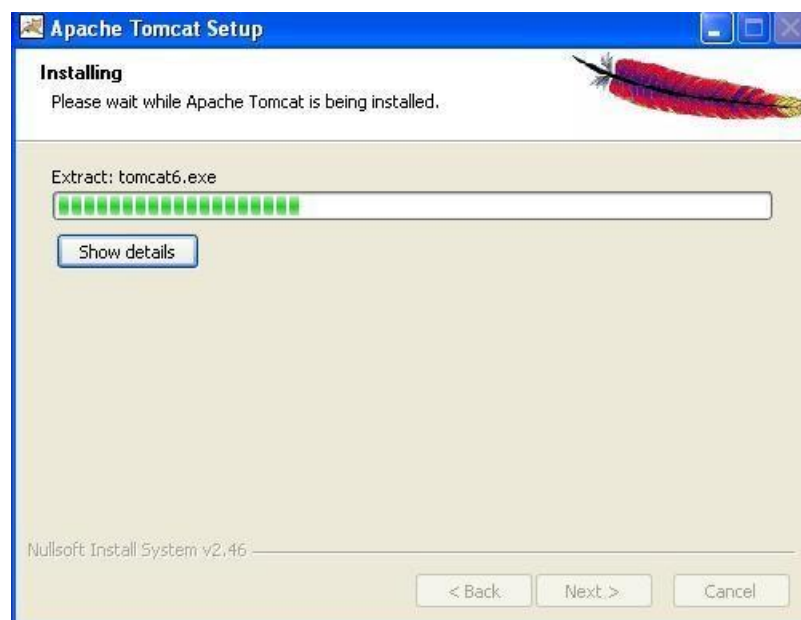
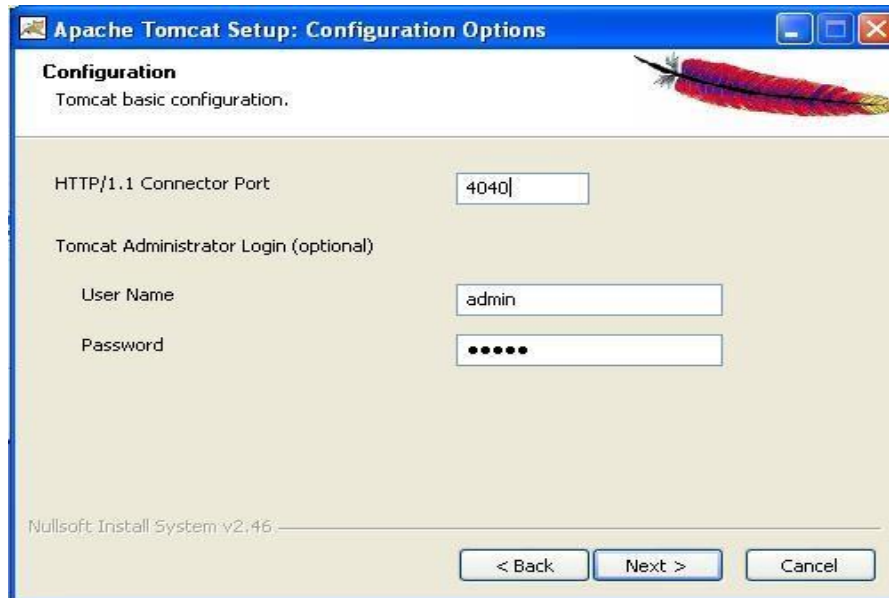
Install TOMCAT web server and APACHE.

While installation assign port number 4040 to TOMCAT and 8080 to APACHE. Make sure that these ports are available i.e., no other process is using this port.

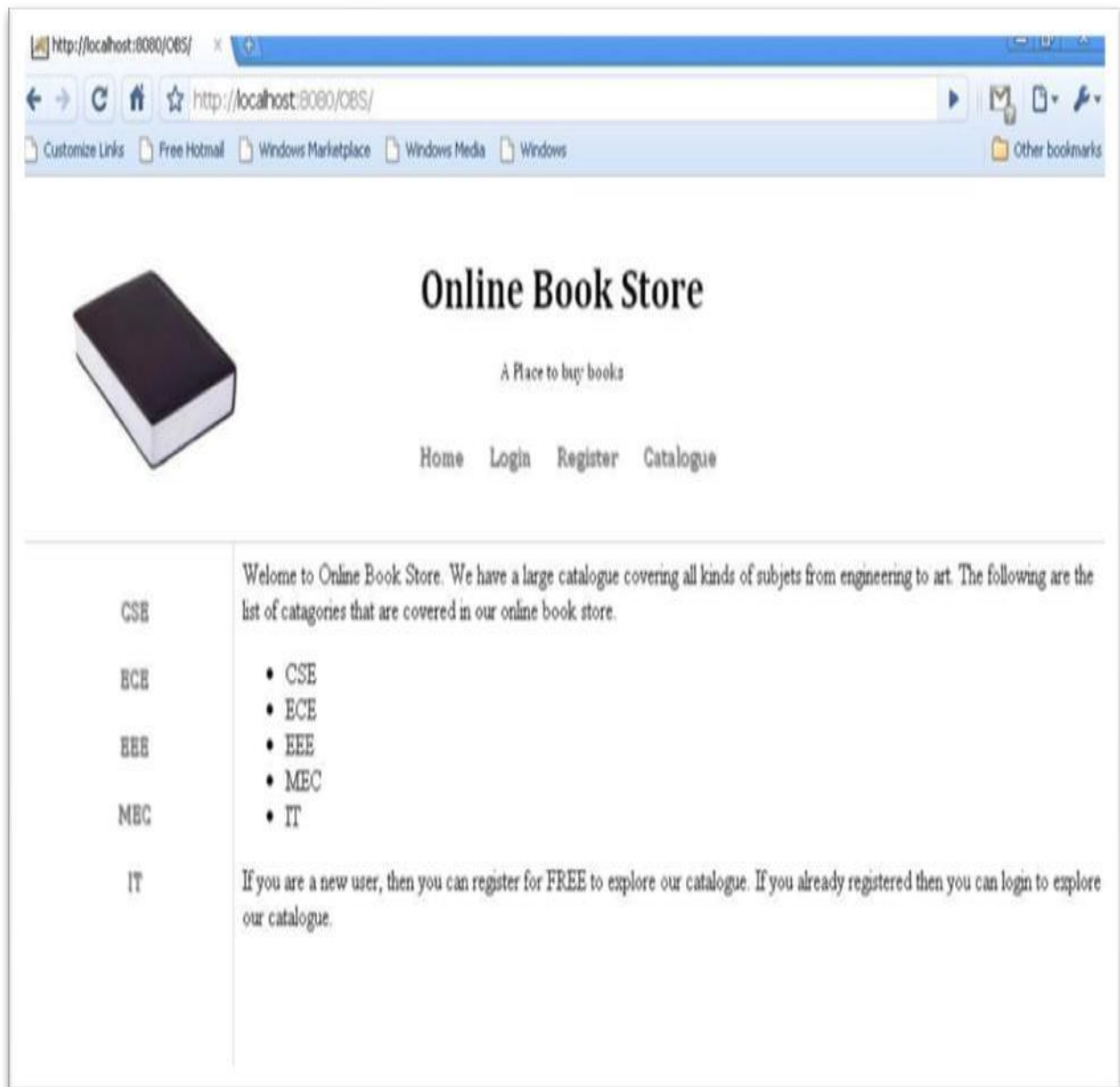
Access the above developed static web pages for books web site, using these servers by putting the web pages developed in week-1 and week-2 in the document root.











The screenshot shows a web browser window with the address bar displaying `http://localhost:8080/OBS/`. The browser's toolbar includes navigation buttons (back, forward, refresh, home), a search bar, and a bookmarks bar with links like 'Customize Links', 'Free Hotmail', 'Windows Marketplace', 'Windows Media', 'Windows', and 'Other bookmarks'.

The web application, titled 'Online Book Store', features a header section with a book icon on the left. The title 'Online Book Store' is centered, followed by the tagline 'A Place to buy books'. Below the tagline are navigation links: 'Home', 'Login', 'Register', and 'Catalogue'.

The main content area is divided into two columns. The left column lists categories: CSE, BCE, EEE, MEC, and IT. The right column contains a welcome message and a list of categories covered in the online book store.

CSE	<p>Welcome to Online Book Store. We have a large catalogue covering all kinds of subjects from engineering to art. The following are the list of categories that are covered in our online book store.</p> <ul style="list-style-type: none"> • CSE • ECE • EEE • MEC • IT
BCE	
EEE	
MEC	
IT	
<p>If you are a new user, then you can register for FREE to explore our catalogue. If you already registered then you can login to explore our catalogue.</p>	

