# WEB PROGRAMMING LABORATORY

| | |
|---|---|
| **Subject Code: 10CSL78** | **I.A. Marks : 25** |
| **Hours/Week : 03** | **Exam Hours: 03** |
| **Total Hours : 42** | **Exam Marks: 50** |

1. Develop and demonstrate a XHTML file that includes Javascript script for the following problems:
   a) Input: A number n obtained using prompt
   Output: The first n Fibonacci numbers
   b) Input: A number n obtained using prompt
   Output: A table of numbers from 1 to n and their squares using **alert**
2. a) Develop and demonstrate, using Javascript script, a XHTML document that collects the USN ( the valid format is: A digit from 1 to 4 followed by two upper-case characters followed by two digits followed by two upper-case characters followed by three digits; no embedded spaces allowed) of the user. Event handler must be included for the form element that collects this information to validate the input. Messages in the alert windows must be produced when errors are detected.
   b) Modify the above program to get the current semester also (restricted to be a number from 1 to 8)
3. a) Develop and demonstrate, using Javascript script, a XHTML document that contains three short paragraphs of text, stacked on top of each other, with only enough of each showing so that the mouse cursor can be placed over some part of them. When the cursor is placed over the exposed part of any paragraph, it should rise to the top to become completely visible.
   b) Modify the above document so that when a paragraph is moved from the top stacking position, it returns to its original position rather than to the bottom.
4. a) Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, Name of the College, Brach, Year of Joining, and e-mail id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.
   b) Create an XSLT style sheet for one student element of the above document and use it to create a display of that element.
5. a) Write a Perl program to display various Server Information like Server Name, Server Software, Server protocol, CGI Revision etc.
   b) Write a Perl program to accept UNIX command from a HTML form and to display the output of the command executed.
6. a) Write a Perl program to accept the User Name and display a greeting message randomly chosen from a list of 4 greeting messages.
   b) Write a Perl program to keep track of the number of visitors visiting the web page and to display this count of visitors, with proper headings.
7. Write a Perl program to display a digital clock which displays the current time of the server.
8. Write a Perl program to insert name and age information entered by the user into a table created using MySQL and to display the current contents of this table.
9. Write a PHP program to store current date-time in a COOKIE and display the 'Last visited on' date-time on the web page upon reopening of the same page.
10. Write a PHP program to store page views count in SESSION, to increment the count on each refresh, and to show the count on web page.
11. Create a XHTML form with Name, Address Line 1, Address Line 2, and E-mail text fields. On submitting, store the values in MySQL table. Retrieve and display the data based on Name.
12. Build a Rails application to accept book information viz. Accession number, title, authors, edition and publisher from a web page and store the information in a database and to search for a book with the title specified by the user and to display the search results with proper headings.

**Note: In the examination *each* student picks one question from the lot of *all* 12 questions.**

# Web Programming Laboratory Manual

## Introduction

This lab is intended to give the students a sound knowledge in the Web side programming. Before going in to the details of the lab, the pre-requisites are the basic knowledge in HTML, XHTML, CSS, XML, JavaScript, Perl, PHP and MySql. Let's look at some of these topics in brief now.

## Basics of HTML: -

**Hyper Text Markup Language (HTML)** is a markup language developed by the W3C people. This can be used as an interface for working our programs. We submit all our requests in the HTML form. It is basically a markup language which describes how the documents are to be formatted.

HTML has two basic entities, the **"Tags"** (Formatting commands) and the strings within the tags called as the **"Directives".** Most of the tags have the following syntax: - <something> that indicates the beginning of the tag and a </something> that indicates the end of the tag.

***NOTE:***

- Tags can either be in lower case or upper case, i.e. there is no difference between <html> and <HTML>
- The order in which parameters of the tag are given is not significant since each of these parameters is named.

## HTML Essentials

An HTML file should be written in the following format and should be saved with .html file extension.

```
<html>
            <head>
                  <title> New Page </title>
            </head>
```

```
        <body>
                 TYPE YOUR TEXT HERE
        </body>
</html>
```

The "*New Page*" title comes on the top of the Browser Window.

## Basic HTML Tags: -

To create a text box

```
        <input type=text name=T1 size=20>
```

To create a Normal Button

```
        <input type=button name=B4 value=GO>
```

To create a Submit Button

```
        <input type=submit name=B1 size=20>
```

To create a Reset Button

```
        <input type=reset name=T1 size=20>
```

To create a Radio Button

```
        <input type=radio value=V1 checked name=R1>
```

To create a Check box

```
        <input type=checkbox name=C1 value=ON >
```

To create a Form

```
        <form method=[GET/POST] action=[url]>

         <input type=submit value=Submit name=B1>

        <input type=reset value=Reset name="B2">

        </form>
```

To create a Text Area

```
<textarea rows=2 name=S1 cols=20></textarea>
```

To create a Drop down Menu

```
<select size=1 name=D1></select>
```

To create a Hyper Link

```
<a href=http://localhost: 8080/a.html>BACK </a>
```

To create a **Marquee** (The Marquee tag ensures that the text scrolls horizontally across the screen. It is usually used by advertisement sites to catch the user's attention

```
<marquee align=middle>Type your text here</marquee>
```

To give Background color

```
<body bgcolor=green>…</body>
```

*(The basics colors can be given literally here. For a more elaborate set if colors, Hex code of the colors can be given. Refer to the possible ranges of the Hex codes in the Text Book)*

## More Miscellaneous Tags: -

1) <h#>……..</h#> - where '#' is a number ranging from 1-6. This is used to set the text size.

2) <pre> - Preformatted text, ensures that the text appears exactly the way it appears in the HTML code thereby preserving the white spaces as well.

3) <br> - Inserts a "New line" character (similar to '\n').

4) To Draw a Horizontal Line (Horizontal Ruler):
   <hr size=4 width="50 %">

5) < b > - Bold, < I > - Italics, <u>-Underline

6) Tables:

    <table>
    <caption> Your Caption here </caption> [Optional Tag]
    <tr>
            <th> Row 1, Col 1 </th> [th implies Table Header]
            <th> Row 2, Col 2 </th>
    </tr>
    <tr>
            <td> Table Definition here </td>
            <td> …………………….. </td>
    </tr>
    </table>

7) Comments:

    < ! - - Your Comments here - - >

8) Background Images:

    <body background = "pathname/abc.gif">
    …………………….
    </body>

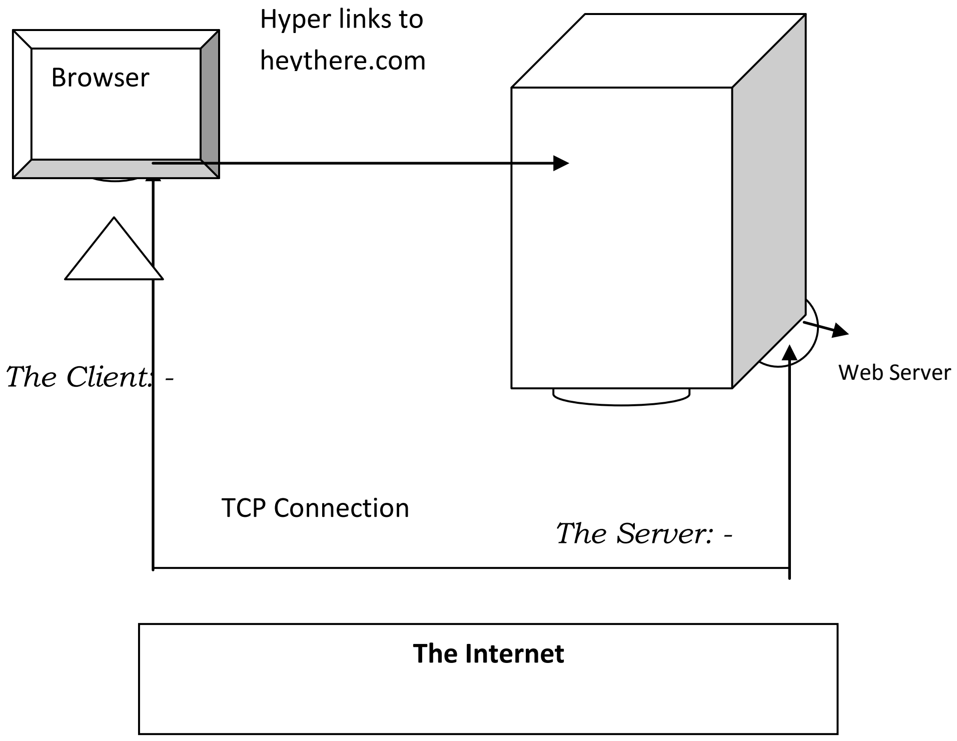
Before we move on further, we need to know how the web exactly works.

## How does the Web work?

The Web is usually accessed through a browser. When the user types in a URL say, **www.mitmysore.in** in the address bar of the browser, the browser makes a socket (Network) connection to the server **www.mitmysore.in**. This name is mapped to an IP address which is of the form 1.2.3.4 by making use of a DNS Server. The browser connects to this server using a logical *port 80*, the port that the server OS opens for internet connections. This is a standard port number.

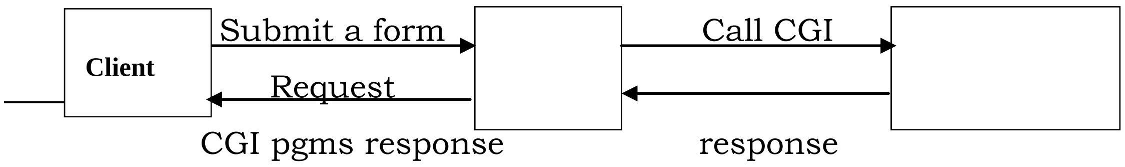
Based on the client request, the server delivers information. The type of data that the server sends back to the client could be a simple plain text (HTML), images, Java Applets etc. this data can be obtained and delivered in three ways.

- *Serving Static Data* – The server does not do any kind of the processing. It merely obtains the data present on its local hard disk and sends it back to the client.
- *Serving Dynamic Data* – The Server does some processing in this case like executing a program and then outputs the result of the program back to the client as a response.
- *Serving Content with Embedded HTML* – Here, an executable code is present with the HTML file. It's not quite static or dynamic.



## Basics of CGI: -

CGI stands for *Common Gateway Interface*. It is a part of the web server that can communicate with other programs running on the server. With CGI, the web server can call up a program, while passing user-specific data to the program. The program then processes that data and the server passes the program's response back to the web browser.



Gateways are programs or scripts used to access information that is not directly readable by the client.

## Basics of Perl: -

Perl is a platform-independent scripting language that stands for *Practical Extraction and Reporting Language*. Perl basically originated as a text processing language and was meant to manage and manipulate a database of text files.

**Essential Features of Perl: -**

1. It is an object-oriented language.
2. Its syntax is C – like.
3. Perl is free format – white space can be scattered about to make the code more readable.
4. All statements must end with a ';'.
5. Variables in Perl do not have to be declared but can be used.
6. Built in functions can be invoked with or without parentheses.
7. Perl scripts are stored as Text files. When executed, the source text file is first compiled into a "*Byte Code*", an intermediate form, not text or binary. Perl then interprets the byte code, executing it.
8. Anything that comes after a '#' symbol is treated as a Comment except the Interpreter line or the Shebang line.

## Basics of MySQL: -

MySQL is an Open source Standard Query Language (SQL) database that is fast, reliable, easy to use and suitable for applications of any size. MySQL can be integrated into Perl programs by using the Perl *DBI (Database Independent Interface)* module. DBI is an API that allows Perl to connect and query a number of SQL Databases such as MYSQL, Oracle, Sybase etc.

For some of the programs in the Lab course, the MySQL database is to be used. For that, the MySQL Server is to be started. The following steps are to be performed in the same sequence on the Linux shell to start the server and create the database along with the table.

**To Start MySQL Server:**

```
# mysql

mysql> create database ise;

mysql> show databases;

mysql> use ise;

mysql> create table student (name varchar(25),age int);

mysql> insert into student values (“e1”,21);

mysql> insert into student values (“e2”,22);

mysql> exit;
```

## About PHP:

PHP is a server-side scripting language. The concept of php is very similar to the JavaScipts or VBScipts. PHP server-side scripting language is similar to JavaScript in many ways, as they both allow you to embed little programs (scripts) into the HTML of a Web page.

**The key difference between JavaScript and PHP:**

The key difference between JavaScript and PHP is that, while the Web browser interprets JavaScript once the Web page containing the script has been downloaded, server-side scripting languages like PHP are interpreted by the Web server before the page is even sent to the browser. Once interpreted, the PHP code is replaced in the Web page by the results of the script, so all the browser sees is a standard HTML file. The script processed entirely by the server. Thus the designation: server-side scripting language.

**HOW TO EMBED PHP INTO HTML:**

Let's look at the example today.php shown below.

```
<html>
<head><title>Today's date</title></head>
<body>
<p>Today's date (according to this web server) is
<?php
Echo( date("l, f ds y.") );
?>
</body>
</html>
```

The above program shows the PHP code embeded in the HTML. Lines between <?php and ?> indicates the php code.

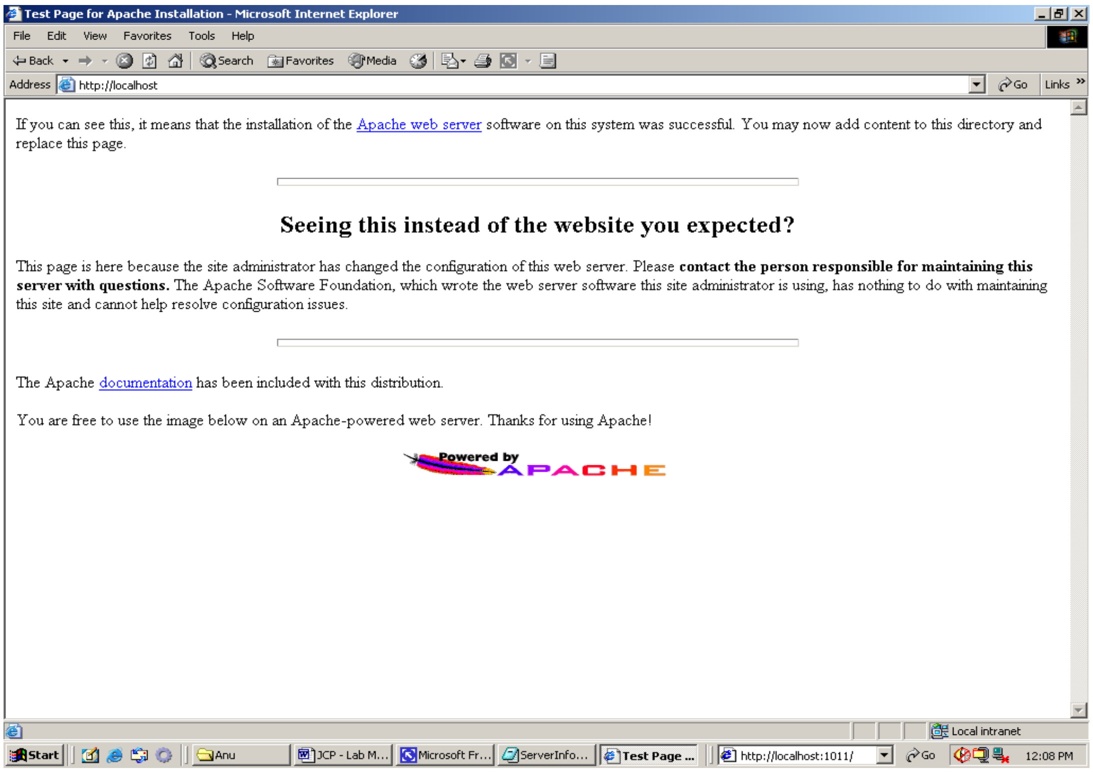
1. <?php means "begin php code".
2. ?> means "end php code".

The Web server is asked to interpret everything between these two delimiters (<?php and ?>) and convert it to regular HTML code before sending the Web page to a
browser that requests it. The browser is presented with something like this:

```
<html>
<head><title>Today's date</title></head>
<body>
<p>today's date (according to this web server) is
Wednesday, june 7th 2000.</p>
</body>
</html>
```

## Apache Http Server:

The web server we are using here is Apache Http Server. It is freely downloadable from the site **www.apache.org**. Once you have downloaded the installer, double click on that and install it in to your system. Then go to start menu □ programs□ Apache HTTP Server□ Control Apache Server. Then click on start to start your server. Then open an Internet explorer and type **http://localhost:80/**. The port number 80 is optional in the URL. If you have successfully installed the server then you will get a screen shown below:


Test Page for Apache Installation - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://localhost
Go Links
Powered by APACHE
Local intranet
Start Anu JCP - Lab M... Microsoft Fr... ServerInfo... Test Page ...
http://localhost:1011/ Go 12:08 PM

If you can see this, it means that the installation of the Apache web server software on this system was successful. You may now add content to this directory and replace this page.

**Seeing this instead of the website you expected?**

This page is here because the site administrator has changed the configuration of this web server. Please **contact the person responsible for maintaining this server with questions.** The Apache Software Foundation, which wrote the web server software this site administrator is using, has nothing to do with maintaining this site and cannot help resolve configuration issues.

The Apache documentation has been included with this distribution.

You are free to use the image below on an Apache-powered web server. Thanks for using Apache!

**1)Develop and demonstrate a XHTML file that includes JavaScript for the following problems:**

**(a)Input: A no. n obtained using prompt.**

**Output: The first n fibonacci numbers.**

**(b)Input: A no. n obtained using prompt**

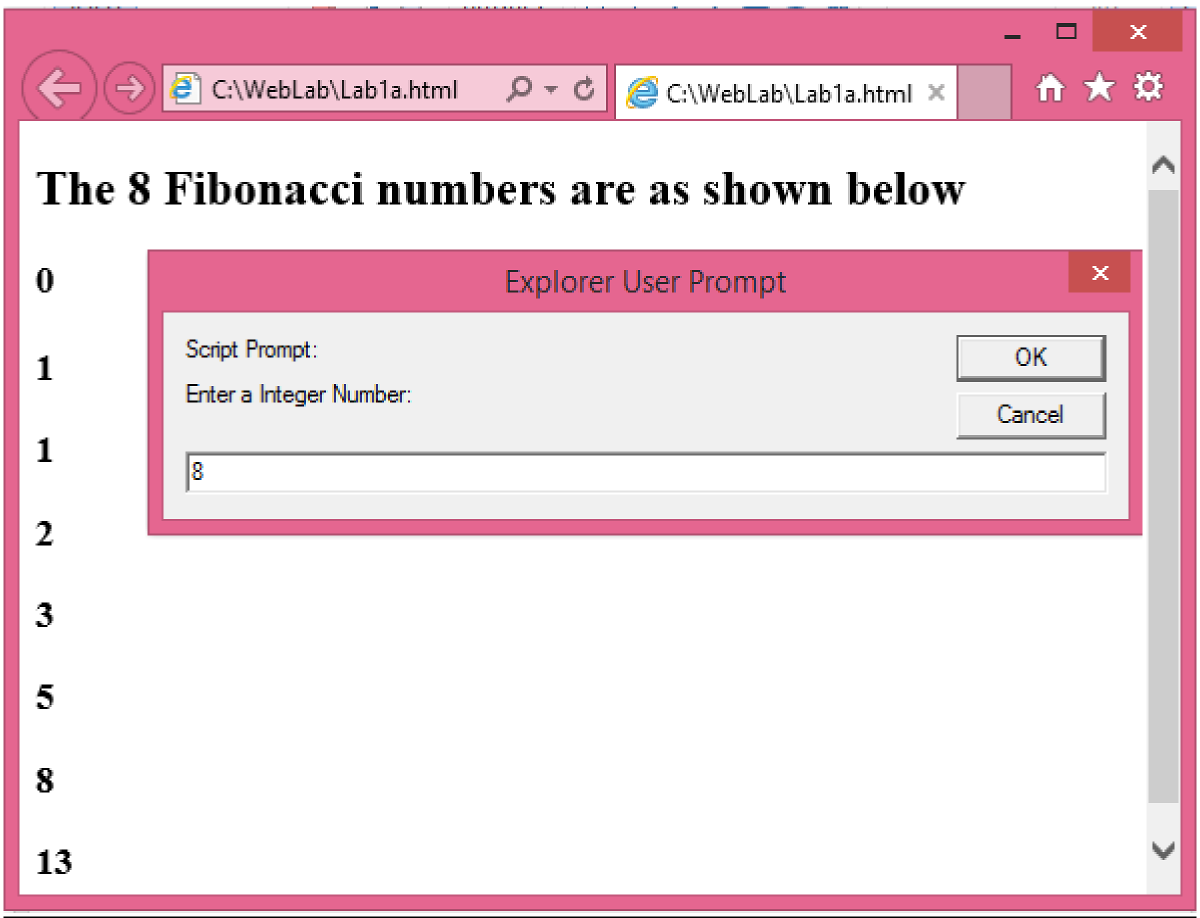
**Output: A table of numbers from 1 to n and their squares using alert.**

## Lab1a.html

```
<?xml version = "1.0" encoding = "utf-8" ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"

"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns = "http://www.w3.org/1999/xhtml">

<body>

<script type="text/javascript">

var fib1=0,fib2=1,fibsum=0;

var n = prompt("Enter a Integer Number : n ", "");

if(n>0)

{

     document.write("<h2>"+"The "+n+" Fibonacci numbers are as
     shown below"+ "</h2>");

     if(n==1)

          document.write("<h3> "+ fib1 + "</h3>");

     else

 document.write("<h3>" + fib1 + "<br/>" + fib2 +  "</h3>");

     for(i=3;i<=n; i++)

     {

          fibsum= fib1 + fib2;

          document.write("<h3> " + fibsum + "</h3>");
```

```
            fib1=fib2;

            fib2=fibsum;

      }

}

else

      alert("Wrong Input!!! Refresh Page and Try again");

</script>

</body></html>
```

**<u>OUTPUT:Lab1a.html</u>**



C:\WebLab\Lab1a.html

**The 8 Fibonacci numbers are as shown below**

Explorer User Prompt

Script Prompt:
Enter a Integer Number:

8

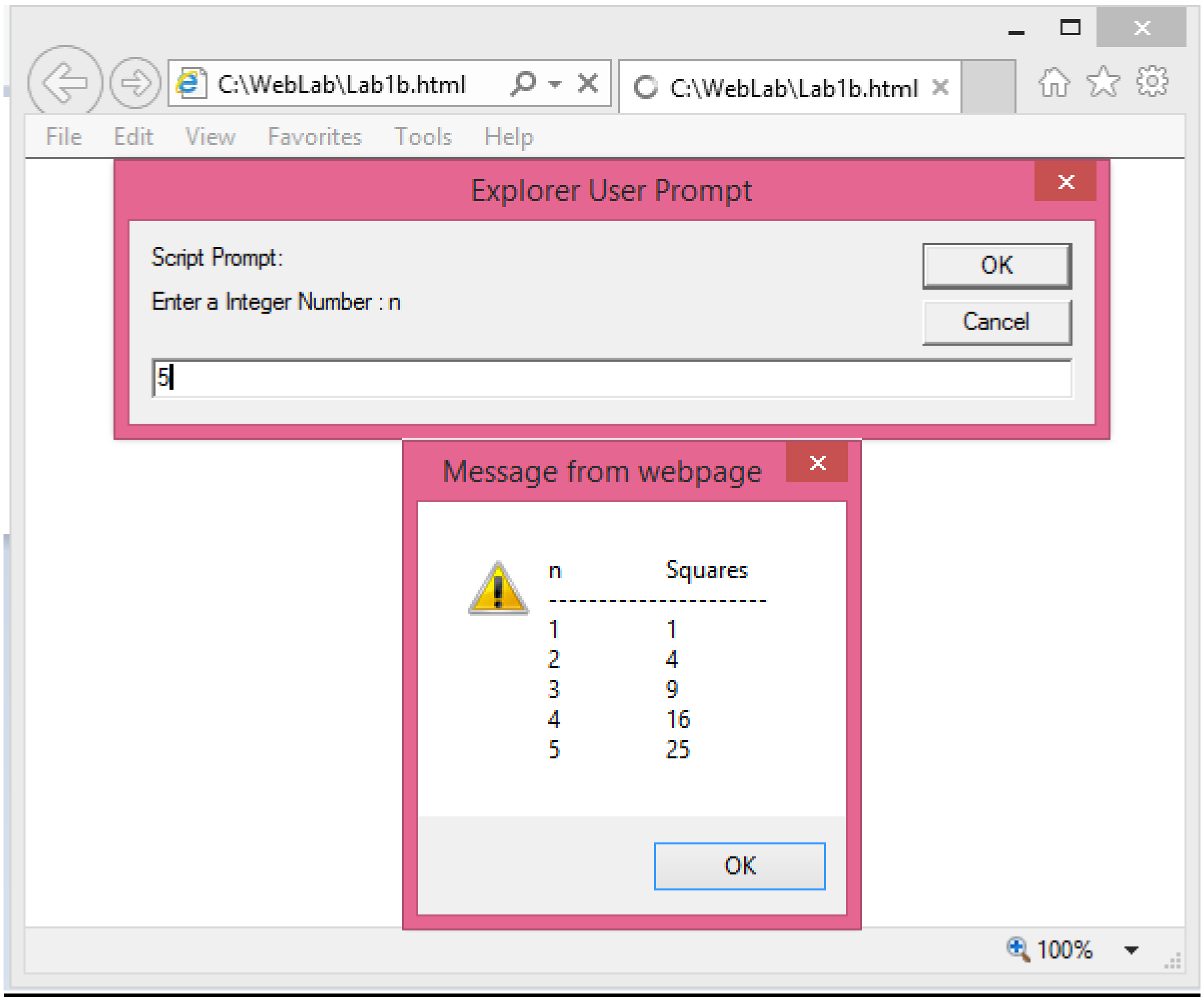
OK
Cancel

```
0

1

1

2

3

5

8

13
```

## Lab1b.html

```
<?xml version = "1.0" encoding = "utf-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<body>  <script type="text/javascript">
var n = prompt("Enter a Integer Number : n ", "");
if(n >0)
{
     var str="n \t Squares \n";
     str=str+"----------------------\n"
     for(i=1; i<=n; i++)
          str = str +i+" \t "+i*i+ "\n";
     alert(str)
}
else
     alert("Wrong Input!!! Refresh page and try again");
</script>
</body></html>
```

**OUTPUT:Lab1b.html**



```
Script Prompt:
Enter a Integer Number : n
5
```

```
n               Squares
-----------------------
1               1
2               4
3               9
4               16
5               25
```

**2)(a)Develop and demonstrate using java script, a XHTML document that collects the USN (the valid format is :a digit from 1 to 4 followed by 2 upper case characters followed by 2 digits followed by 2 upper case characters followed by 3 digits. No embedded spaces allowed) of the user. Event handler must be included for the form element that collects this information to validate the input. Messages in the alert windows must be produced when errors are detected.**

**(b) Modify the above program to get the current semester also (Restricted to be a no. from 1 to 8).**

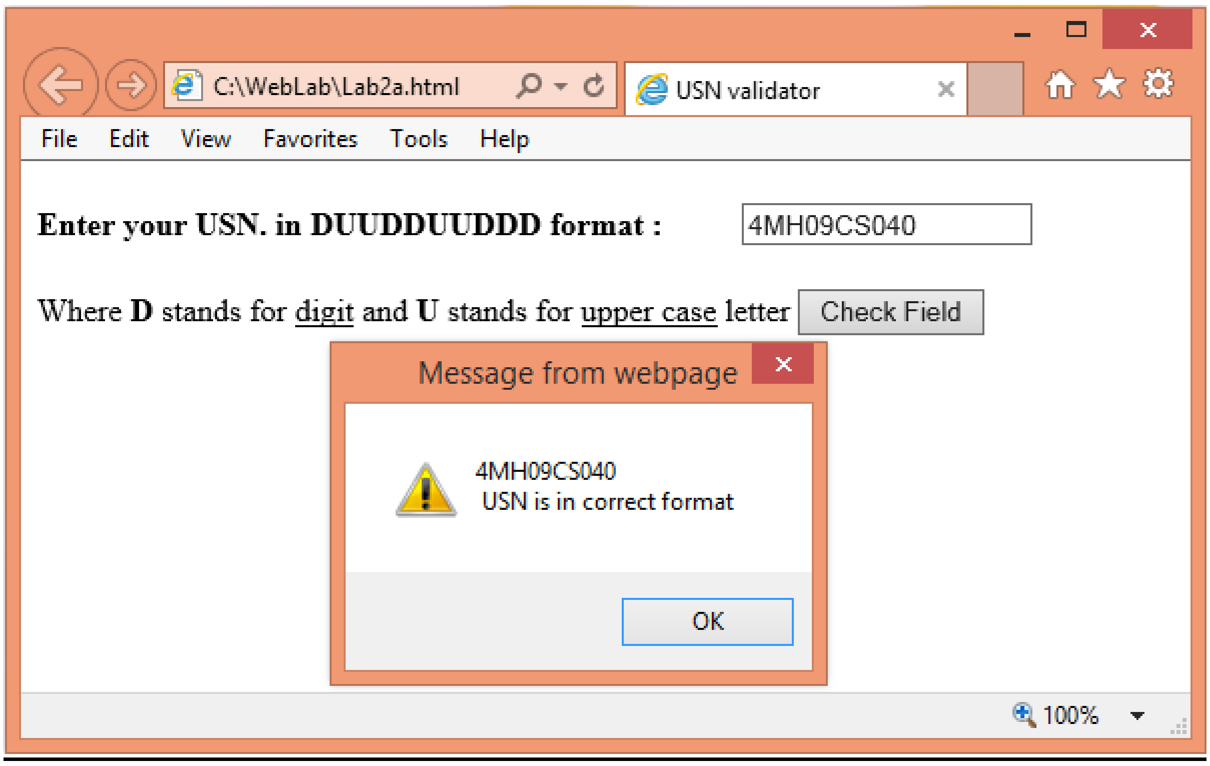
## Lab2a.html

```
<?xml version = "1.0" encoding = "utf-8" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns = "http://www.w3.org/1999/xhtml">
<script type="text/javascript">
var usn,regexp;

function formValidator()
{
     usn = document.getElementById("field1");
 regexp=/[1-4][A-Z][A-Z][0-9][0-9][A-Z][A-Z][0-9][0-9][0-9]$/;
     if(usn.value.length == 0)
     {
          alert("USN is empty");
          return;
     }

else if(!usn.value.match(regexp))
```

```
     {

          alert(usn.value+"\n Entered USN is Wrong!!!,\n It
          should be in DUUDDUUDDD format");

          return;

     }

     alert(usn.value+"\n USN is in correct format");

}

</script>

<head><title>USN validator</title></head>

<body>

<form onsubmit="return formValidator()">

<h4>Enter your USN. in DUUDDUUDDD format : <input type="text"
id="field1"/></h4>

Where D stands for digit and U stands for upper case letter

<input type="submit" value="Check Field" />

</form>

</body>

</html>
```

**OUTPUT:Lab2a.html**

**Run 1:**

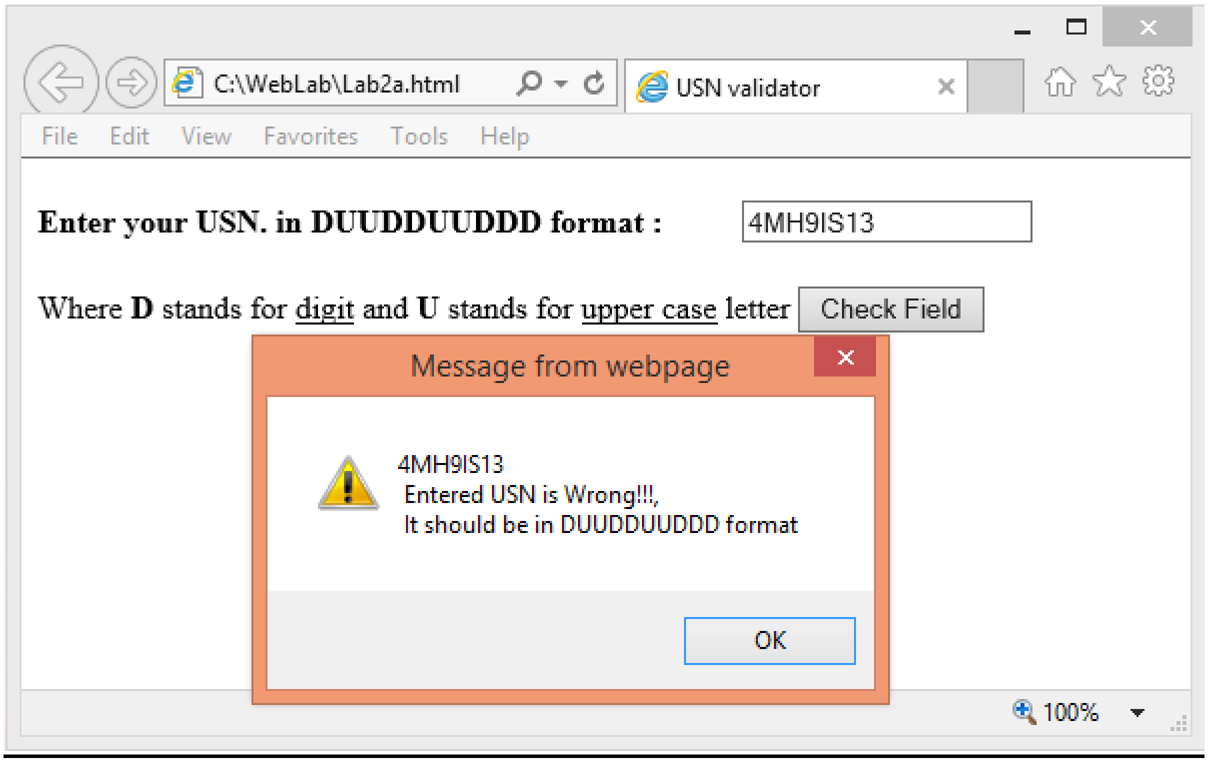


Enter your USN. in DUUDDUUDDD format : 4MH09CS040

Where D stands for digit and U stands for upper case letter

4MH09CS040
USN is in correct format

**Run 2:**



Enter your USN. in DUUDDUUDDD format : 4MH9IS13

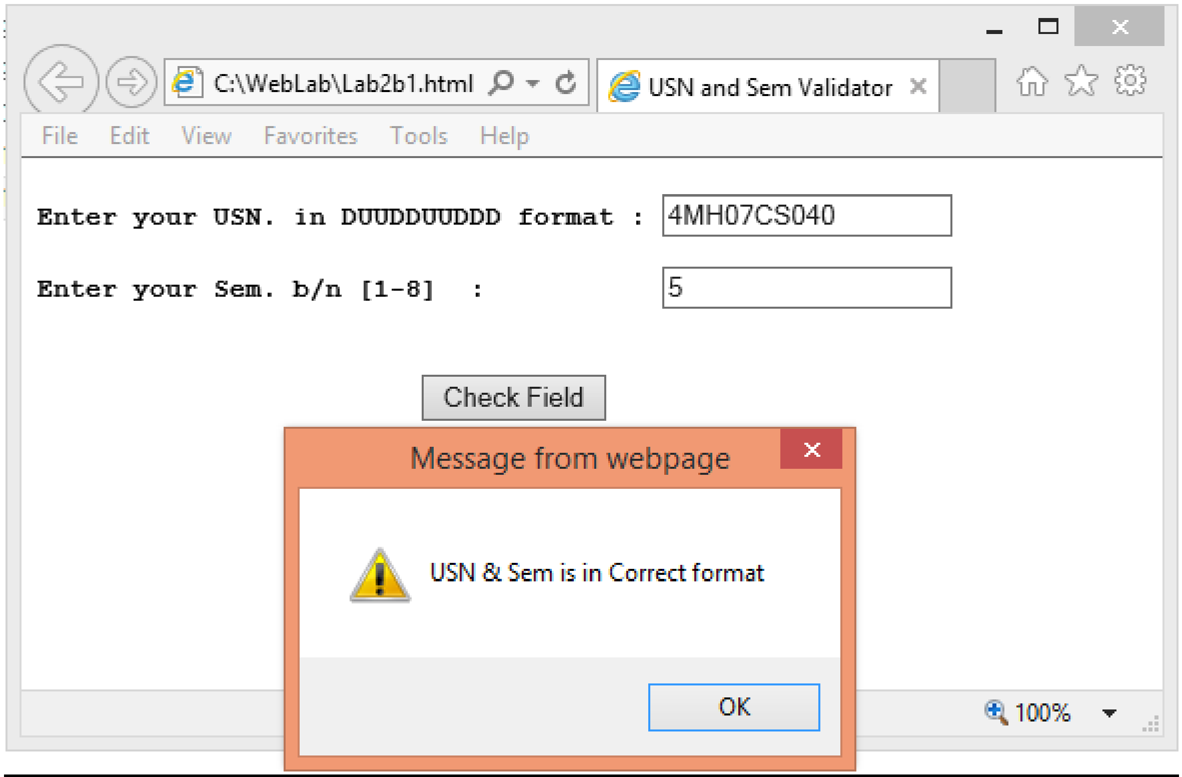
Where D stands for digit and U stands for upper case letter

4MH9IS13
Entered USN is Wrong!!!,
It should be in DUUDDUUDDD format

## Lab2b.html

```
<?xml version = "1.0" encoding = "utf-8" ?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"

"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns = "http://www.w3.org/1999/xhtml">

<script type="text/javascript">

function formValidator()

{

     var usn = document.getElementById("field1");

     var sem = document.getElementById("field2");

     var regexp1=/[1-4][A-Z][A-Z][0-9][0-9][A-Z][A-Z][0-9][0-
     9][0-9]$/;

     var regexp2 = /[1-8]$/;

     if(usn.value.length == 0||sem.value.length==0)

     {

          alert("Fields are empty");

          return;

     }

     if(usn.value.match(regexp1)&&sem.value.match(regexp2))

     {

          alert("USN & Sem is in Correct format");

          return;

     }

     if(!usn.value.match(regexp1))

     {

          alert("Entered USN is Wrong!!!\n It should be in
          DUUDDUUDDD format");

          return;

     }
     if(!sem.value.match(regexp2))
     {
          alert("Invalid Semester Number,Enter it again ");
          return;
     }
}
</script>
<head><title>USN and Sem Validator</title></head>
<body>
<form onsubmit="return formValidator()">
<h4>Enter your USN. in DUUDDUUDDD format :
<input type="text" id="field1"/> <BR/>
Enter your Sem. b/n [1-8]  :
<input type="text" id="field2"/> <BR/></h4>
<input type="submit" value="Check Field" />
</form>
</body>
</html>
```

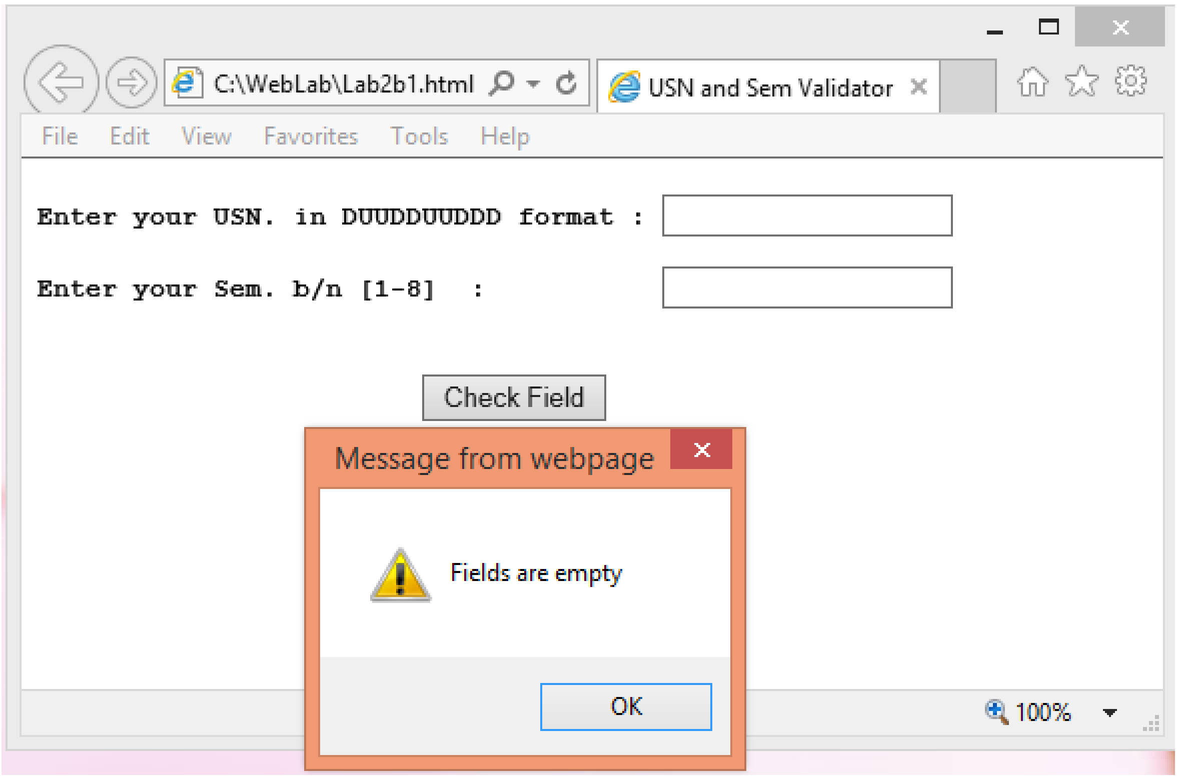
**OUTPUT:Lab2b.html**



Enter your USN. in DUUDDUUDDD format : 4MH07CS040

Enter your Sem. b/n [1-8] : 5

USN & Sem is in Correct format



Enter your USN. in DUUDDUUDDD format :

Enter your Sem. b/n [1-8] :

Fields are empty

**3) Develop and demonstrate, using javascript script, a XHTML document that contains 3 short paragraphs of text, stacked on top of each other, with only enough of each showing so that the mouse cursor can be placed over some part of them. When the cursor is placed over exposed part of any paragraph, it should rise to the top to become completely visible.**

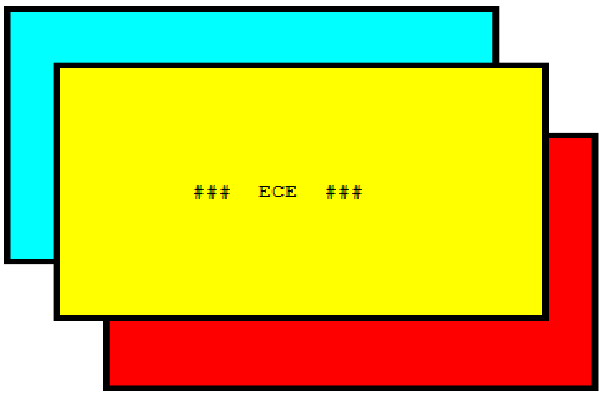
**b) Modify the above document so that when a paragraph is moved from the top stacking position, it returns to its original position rather than to the bottom**

## Lab3a.html

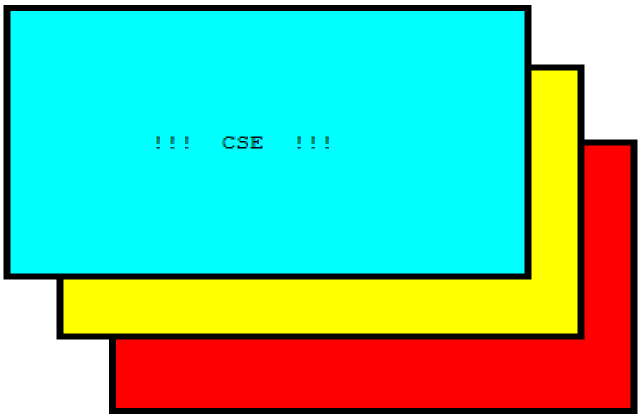
```
<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE html PUBIC "-//W3C//DTD XHTML 1.1//EN"

"http://www.w3.org/TR?xhtml11/DTD/xhtml11.dtd">

<html>

<script type="text/javascript">

var stack1="stack1";


function toTop(curStack)

{

        var oldStack=document.getElementById(stack1).style;

        oldStack.zIndex="0";

        var newStack=document.getElementById(curStack).style;

        newStack.zIndex="10";

        stack1=curStack;

}

</script>

<style type="text/css">



.para1
```

```
{
     position:absolute;top:30%;left:30%;z-index:0; width:200;
     border:solid;padding:100;background-color:aqua;
}
.para2
{
     position:absolute;top:35%;left:35%;z-index:0; width:200;
     border:solid; padding:100; background-color:yellow;
}
.para3
{
     position:absolute;top:40%;left:40%;z-index:0;width:200;
     border:solid; padding:100; background-color:red;
}
</style>
<body>
<p class="para1" id="stack1"  onmouseover="toTop('stack1')">
     !!!  CSE  !!! </p>
<p class="para2" id="stack2"  onmouseover="toTop('stack2')">
***  ISE  *** </p>
<p class="para3" id="stack3"  onmouseover="toTop('stack3')">
###  ECE  ### </p>
</body>
</html>
```

**OUTPUT: Lab3a.html:** Stacking element shown without placing the mouse cursor.



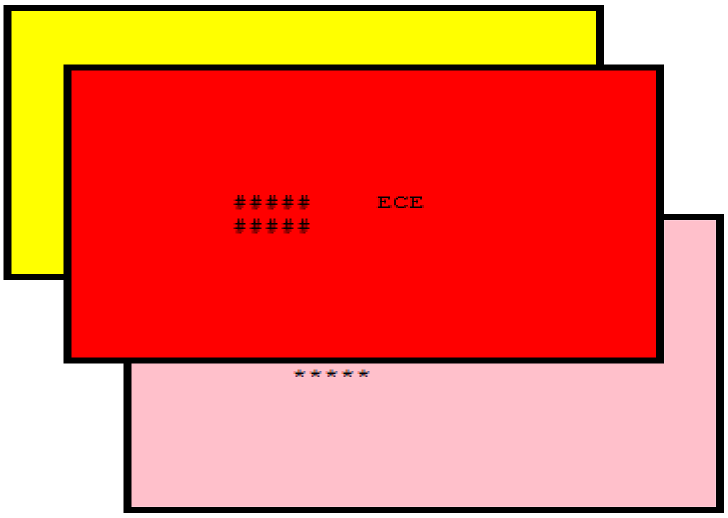
After placing the mouse cursor over an element.



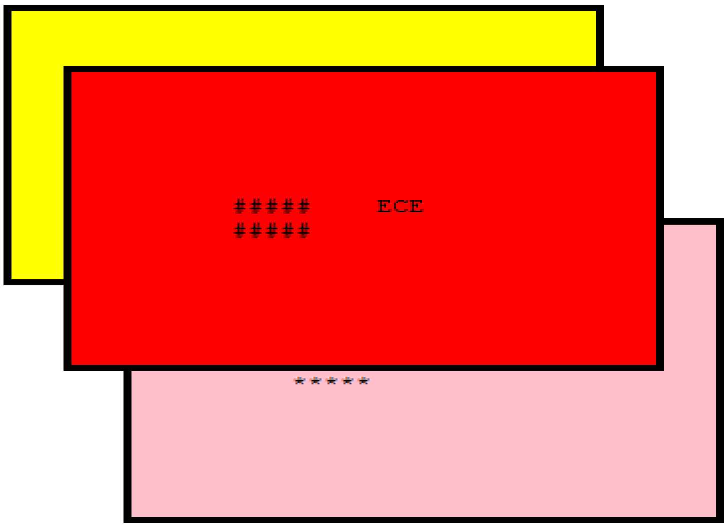
## Lab3b.html

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR?xhtml11/DTD/xhtml11.dtd">
<html>
<script type="text/javascript">
var stack1="stack1";
function toTop(curStack)
{
        var oldStack=document.getElementById(stack1).style;
        oldStack.zIndex="0";
        var newStack=document.getElementById(curStack).style;
        newStack.zIndex="10";

        stack1=curStack;

}

</script>

<style type="text/css">

.para1

{

     position:absolute;top:30%;left:30%;z-index:0; width:200;

     border:solid;padding:100;background-color:aqua;

}

.para2

{

     position:absolute;top:35%;left:35%;z-index:0;width:200;

     border:solid; padding:100; background-color:yellow;

}

.para3

{

     position:absolute;top:40%;left:40%;z-index:0;width:200;

     border:solid; padding:100; background-color:red;

}

</style>

<body>

<p class="para1" id="stack1"  onmouseover="toTop('stack1')"
onmouseout="toTop('stack3')">    !!!  CSE  !!! </p>

<p class="para2" id="stack2"  onmouseover="toTop('stack2')"
onmouseout="toTop('stack3')">    ***  ISE  *** </p>

<p class="para3" id="stack3"  onmouseover="toTop('stack3')"
onmouseout="toTop('stack3')">    ###  ECE  ### </p>

</body>

</html>
```

**OUTPUT:Lab3b.html :** Stacking elements without placing the mouse cursor



Even after placing the cursor on the element the stacked elements placed to their original place



**4a) Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include USN, Name, Name of the College, Brach, Year of Joining, and e-mail id. Make up sample data for 3 students. Create a CSS style sheet and use it to display the document.**

**b) Create an XSLT style sheet for one student element of the above document and use it to create a display of that element.**

## Lab4a.xml

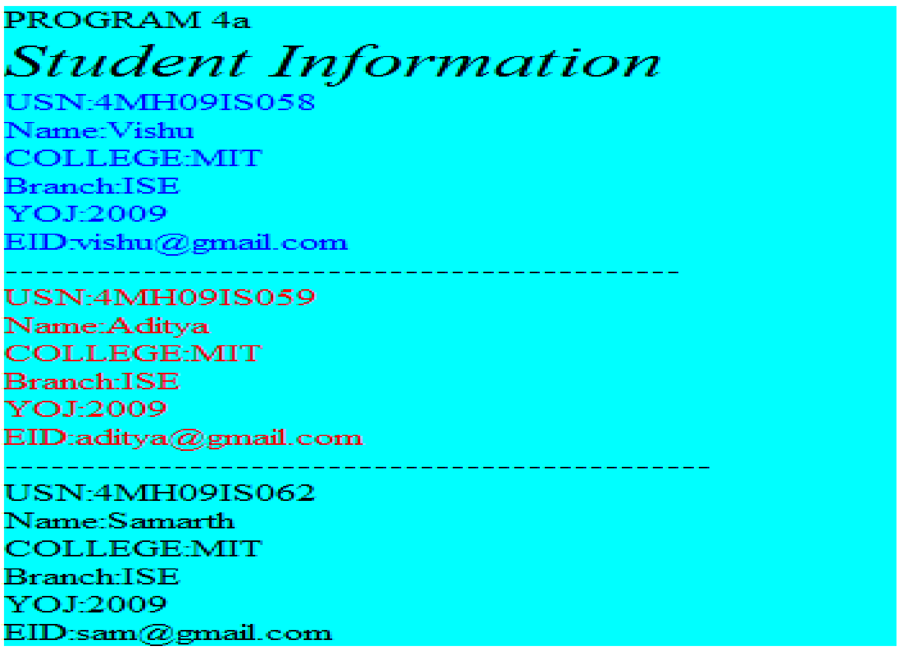
```
<?xml version="1.0" encoding="utf-8" ?>

<?xml-stylesheet type="text/css" href="Lab4a.css" ?>

<student>PROGRAM 4a

<stud-info>Student Information</stud-info>

  <stud1>

         <usn>USN:4MH09IS058</usn>

         <name>Name:Vishu</name>

         <noc>COLLEGE:MIT</noc>

         <branch>Branch:ISE</branch>

         <yoj>YOJ:2009</yoj>

         <eid>EID:vishu@gmail.com</eid>

  </stud1>

--------------------------------------------

  <stud2>

         <usn>USN:4MH09IS059</usn>

         <name>Name:Aditya</name>

         <noc>COLLEGE:MIT</noc>

         <branch>Branch:ISE</branch>

         <yoj>YOJ:2009</yoj>

         <eid>EID:aditya@gmail.com</eid>

  </stud2>

----------------------------------------------

  <stud3>
```

```
          <usn>USN:4MH09IS062</usn>

          <name>Name:Samarth</name>

          <noc>COLLEGE:MIT</noc>

          <branch>Branch:ISE</branch>

          <yoj>YOJ:2009</yoj>

          <eid>EID:sam@gmail.com</eid>

  </stud3>

</student>
```

## Lab4a.css

```
stud-info

{

     display:block; font-style:italic; font-size:200%;

}

student

{

     display : block; background-color:aqua;font-size:100%;

}

stud1

{

     display : block; color:blue;

}

stud2

{
     display : block; color:red;

}

stud3

{

     display : block; color:black;

}

usn,name,noc,branch,yoj,eid

{    display : block;    }
```

## OUTPUT: Lab4a.xml



```
PROGRAM 4a
Student Information
USN:4MH09IS058
Name:Vishu
COLLEGE:MIT
Branch:ISE
YOJ:2009
EID:vishu@gmail.com
-----------------------------------------------
USN:4MH09IS059
Name:Aditya
COLLEGE:MIT
Branch:ISE
YOJ:2009
EID:aditya@gmail.com
-------------------------------------------------
USN:4MH09IS062
Name:Samarth
COLLEGE:MIT
Branch:ISE
YOJ:2009
EID:sam@gmail.com
```

## Lab4b.xml

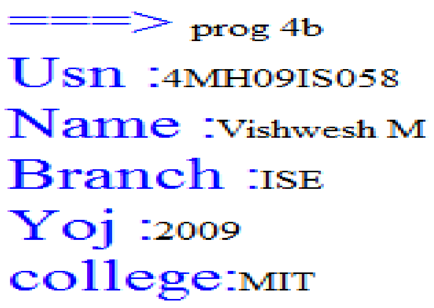
```
<?xml version="1.0" encoding="utf-8" ?>

<?xml-stylesheet type="text/xsl" href="Lab6b.xsl" ?>

<stud>

       <title>prog 6b</title>

       <usn>4MH09IS058</usn>
       <name>Vishwesh M</name>

       <coll>MIT</coll>

       <branch>ISE</branch>

       <yoj>2009</yoj>

       <eid>vishu363@gmail.com</eid>

</stud>
```

## Lab4b.xsl

```
<?xml version="1.0" encoding="utf-8"?>

<xsl:stylesheet version="1.0"

xmlns:xsl="http://www.w3.org/1999/XSL/Transform"

xmlns="http://www.w3.org/1999/xhtml">

<xsl:template match="/">

<span style="font-size:20pt;color:blue"> ===> </span>

<span><xsl:value-of select="stud/title" /><br /></span>

<span style="font-size:20pt;color:blue">Usn    :</span>

<span><xsl:value-of select="stud/usn" /><br /></span>

<span style="font-size:20pt;color:blue">Name   :</span>

<span><xsl:value-of select="stud/name" /><br /></span>

<span style="font-size:20pt;color:blue">Branch :</span>

<span><xsl:value-of select="stud/branch" /><br /></span>

<span style="font-size:20pt;color:blue">Yoj    :</span>

<span><xsl:value-of select="stud/yoj" /><br /></span>

<span style="font-size:20pt;color:blue">college:</span>
```

```
<span><xsl:value-of select="stud/coll" /><br /></span>

</xsl:template>

</xsl:stylesheet>
```

**OUTPUT: Lab4b.xml**



```
===> prog 4b
Usn :4MH09IS058
Name :Vishwesh M
Branch :ISE
Yoj :2009
college:MIT
```

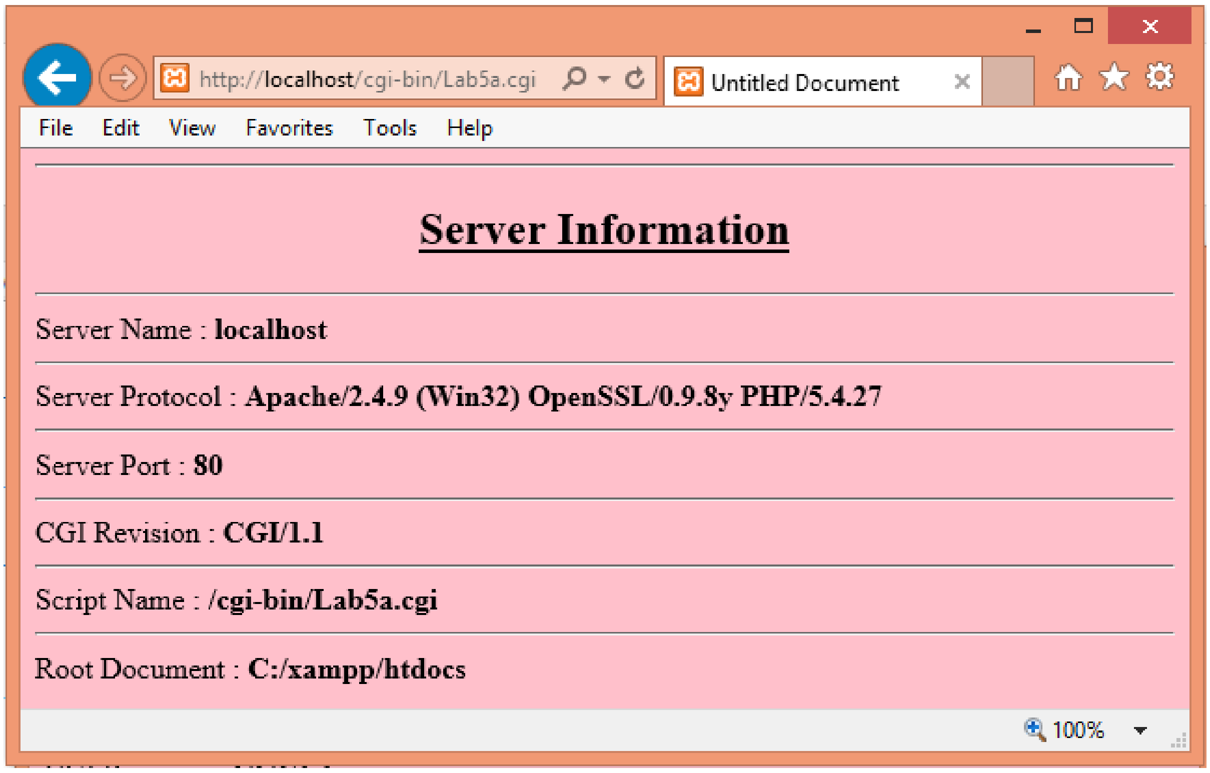
**5a) Write a Perl program to display various server information like Server Name, Server Software, Server protocol, CGI Revision etc.**

To display the server information, five environment variables, SERVER_NAME, SERVER_PORT, SERVER_SOFTWARE, SERVER_PROTOCOL, and CGI_REVISION are used to print the name of the machine where the server is running, the port the server is running on, the server software, and the CGI revisions.

## Perl Script: Lab5a.cgi

```
#!"C:\xampp\perl\bin\perl.exe"

use strict;

use CGI':standard';

print

header(),

start_html(-bgcolor=>"pink"),

hr(),

h2("Server Information"),

hr(),

"Server Name : $ENV{SERVER_NAME}",

hr(),

"Server Protocol : $ENV{SERVER_SOFTWARE}",

hr(),

"Server Port : $ENV{SERVER_PORT}",

hr(),

"CGI Revision : $ENV{GATEWAY_INTERFACE}",

hr(),

"Script Name : $ENV{SCRIPT_NAME}",

hr(),

"Root Document : $ENV{DOCUMENT_ROOT}",

end_html();
```

**OUTPUT: lab5a.cgi**


http://localhost/cgi-bin/Lab5a.cgi
Untitled Document
File Edit View Favorites Tools Help
100%

Server Information

Server Name : **localhost**

Server Protocol : **Apache/2.4.9 (Win32) OpenSSL/0.9.8y PHP/5.4.27**

Server Port : **80**

CGI Revision : **CGI/1.1**

Script Name : **/cgi-bin/Lab5a.cgi**

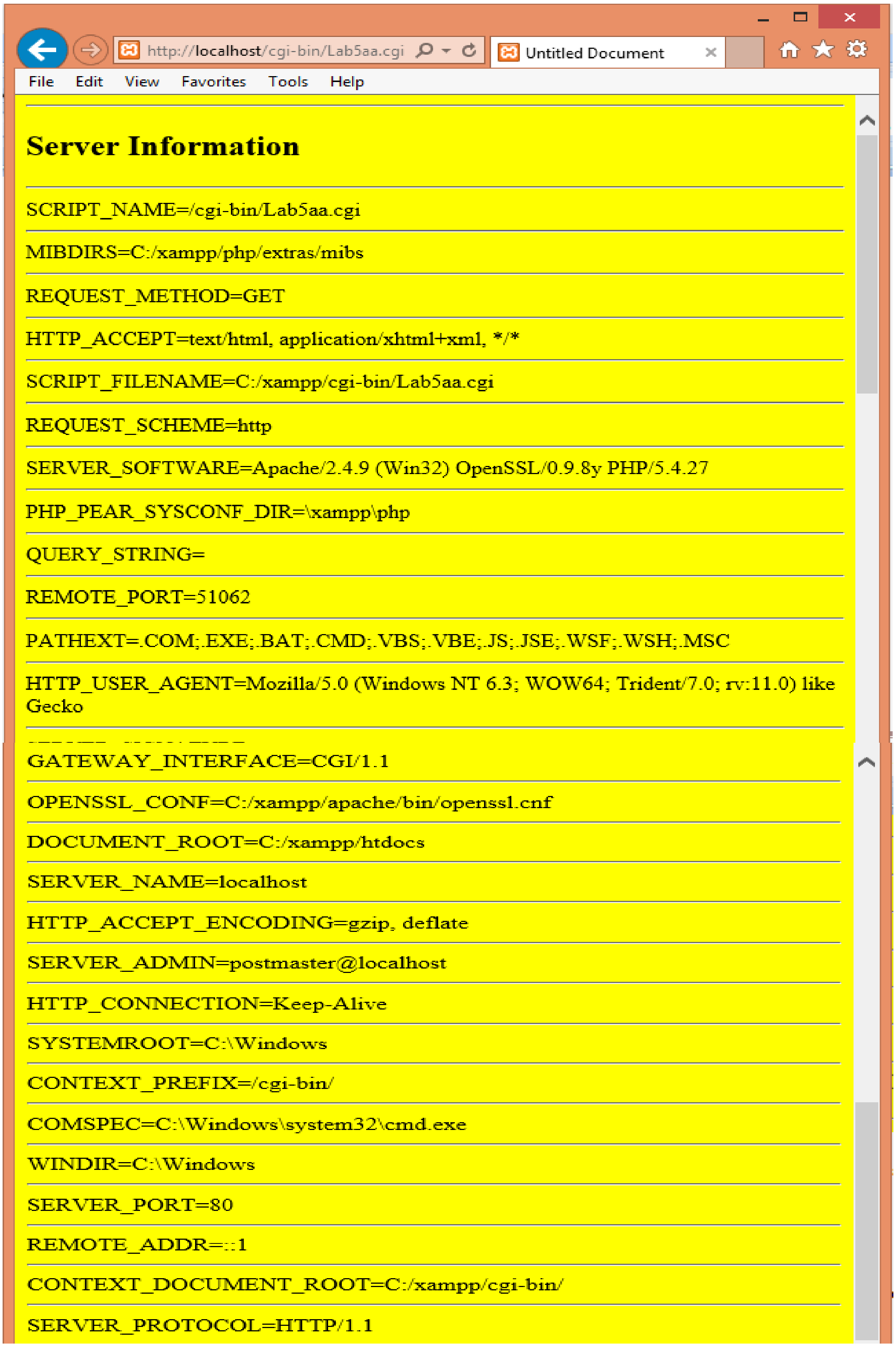
Root Document : **C:/xampp/htdocs**

**OR**

**Perl Script: Lab5aa.cgi**

```
#!"C:\xampp\perl\bin\perl.exe"

use strict;

use CGI':standard';

print

header(),

start_html(-bgcolor=>"yellow"),

hr(),h2("Server Information"),hr();

foreach my $parm(keys(%ENV))

{

     print "$parm=$ENV{$parm}";

     print "<hr>";

}

print end_html();
```

**OUTPUT: lab5aa.cgi**


http://localhost/cgi-bin/Lab5aa.cgi
Untitled Document
File Edit View Favorites Tools Help

```
Server Information

SCRIPT_NAME=/cgi-bin/Lab5aa.cgi
MIBDIRS=C:/xampp/php/extras/mibs
REQUEST_METHOD=GET
HTTP_ACCEPT=text/html, application/xhtml+xml, */*
SCRIPT_FILENAME=C:/xampp/cgi-bin/Lab5aa.cgi
REQUEST_SCHEME=http
SERVER_SOFTWARE=Apache/2.4.9 (Win32) OpenSSL/0.9.8y PHP/5.4.27
PHP_PEAR_SYSCONF_DIR=\xampp\php
QUERY_STRING=
REMOTE_PORT=51062
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
HTTP_USER_AGENT=Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; rv:11.0) like
Gecko
GATEWAY_INTERFACE=CGI/1.1
OPENSSL_CONF=C:/xampp/apache/bin/openssl.cnf
DOCUMENT_ROOT=C:/xampp/htdocs
SERVER_NAME=localhost
HTTP_ACCEPT_ENCODING=gzip, deflate
SERVER_ADMIN=postmaster@localhost
HTTP_CONNECTION=Keep-Alive
SYSTEMROOT=C:\Windows
CONTEXT_PREFIX=/cgi-bin/
COMSPEC=C:\Windows\system32\cmd.exe
WINDIR=C:\Windows
SERVER_PORT=80
REMOTE_ADDR=::1
CONTEXT_DOCUMENT_ROOT=C:/xampp/cgi-bin/
SERVER_PROTOCOL=HTTP/1.1
```

**5b) Write a Perl program to accept UNIX command from a HTML form and to display the output of the command executed.**

To display the Unix command from a HTML form. Program includes:

  The HTML Page provides a simple interface to accept the UNIX command from the user. After the command is entered, the Perl program is invoked. The command entered by the user is sent via the Query String .
  The Perl program has a “use CGI ‘: standard’”. This “use” pragma ensures that the program has access to all the methods that CGI.pm (Perl Module) provides. The “standard” signifies that only appropriate functions are used for the wide range of browsers. An alternate of this could be “use CGI ‘: all’”
  The param() function is used to capture the parameters that are received from the HTML page. This value is stored in a variable called “$comm”. To execute the command, the back ticks (`) are used. An alternate approach to this would be to use the system ( ) function

## Perl Script: lab5b.cgi

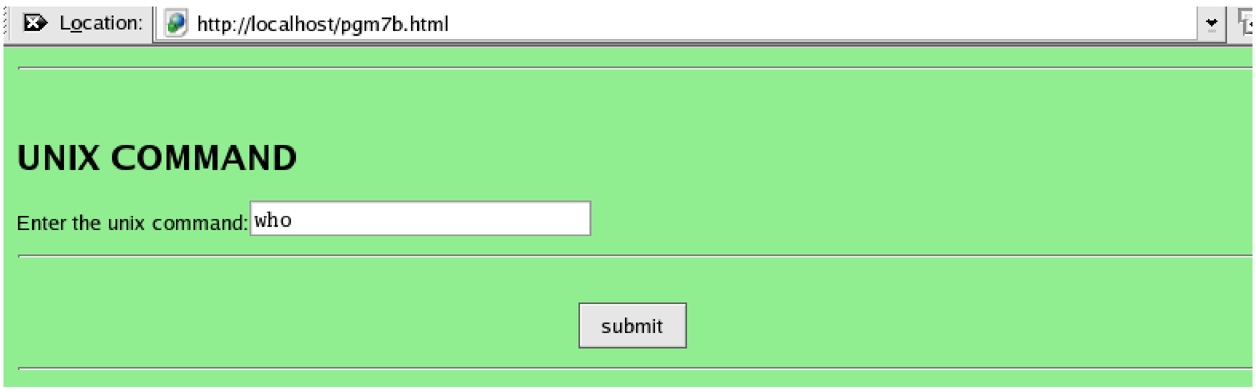
```
#!"C:\xampp\perl\bin\perl.exe"
use CGI':standard';

print 〝content-type: text/html \n\n 〞 ;

$c=param('com');

system($c);

exit(0);
```

## Html : Lab5b.html

```
<html><body bgcolor = 〞pink〞>

<form action = 〞/cgi-bin/lab7b.cgi〞 >

<input type= 〞text〞 name= 〝com〞 >

<input type= 〞submit〞  value= 〝submit〞>

</form>
```

```
</body>

</html>
```

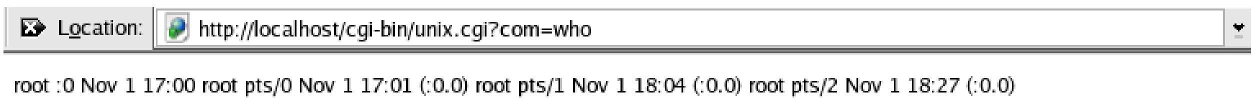
## **OUTPUT: Lab5b.html**



UNIX COMMAND

Enter the unix command: who

submit



```
root :0 Nov 1 17:00 root pts/0 Nov 1 17:01 (:0.0) root pts/1 Nov 1 18:04 (:0.0) root pts/2 Nov 1 18:27 (:0.0)
```

**6a) Write a Perl program to accept the username and display greeting message randomly chosen from a list of greeting messages.**

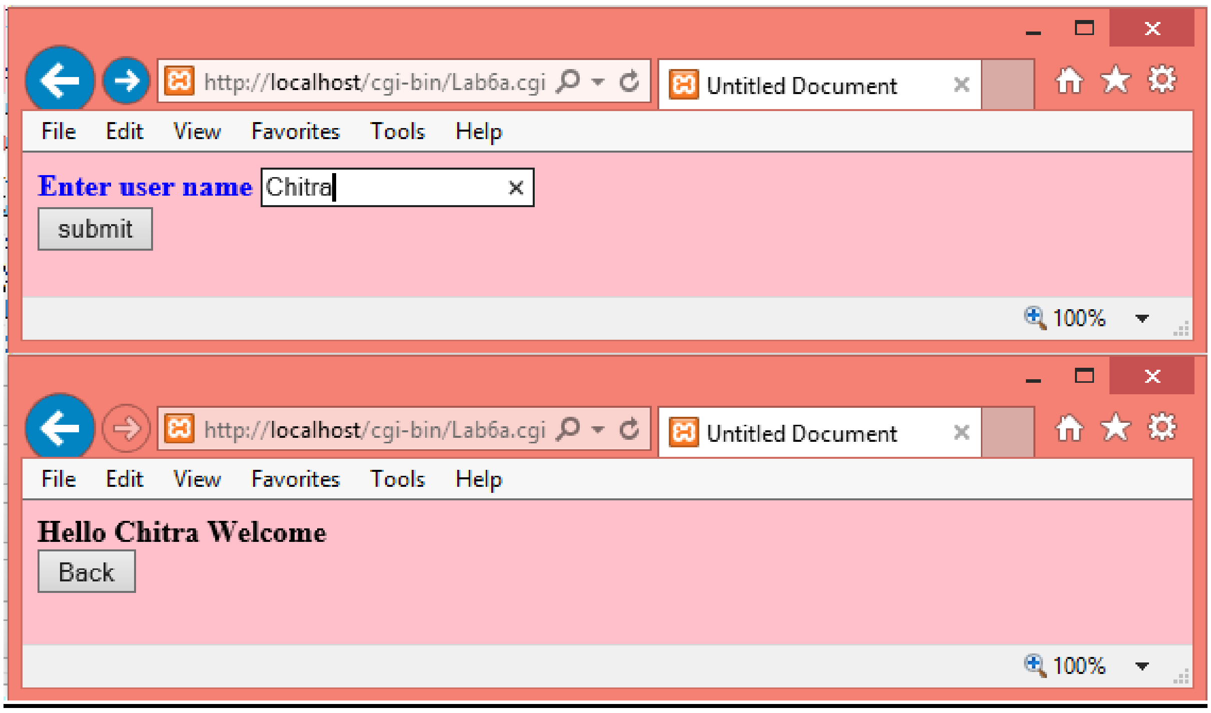
In this we have embedded an html page to a perl to enter user name in the form and click on “Submit”. The name entered by the user is then used by the Perl program using the param ( ) function. The greeting “Hello <*name*>” is printed as the output.

## Perl Script: Lab6a.cgi

```
#!"C:\xampp\perl\bin\perl.exe"

use CGI':standard';

@msgs=("Welcome " ,"have a nice day","hi","how are you");

$len=@msgs.length;

$n = int(rand($len));

if (param)

{

     print header( );

     print start_html(-bgcolor=>"pink");

     $name=param("name");

     print b("Hello $name $msgs[$n]"),br();

     print start_form();

     print submit(-value=>"Back");

     print end_form();

     print end_html();

}

else

{

     print header();

     print start_html(-bgcolor=>"pink", -text =>"blue");
```

```
    print start_form();

    print b("Enter user name ");

    print textfield(-name=>"name"),br();

    print submit(-value =>"submit");

    print end_form();

    print end_html();

}
```

## OUTPUT: lab6a.cgi


http://localhost/cgi-bin/Lab6a.cgi
Untitled Document
File Edit View Favorites Tools Help
100%

Enter user name Chitra
submit

Hello Chitra Welcome
Back

**6b) Write a Perl program to keep track of number of visitors visiting the web page and to display this count of visitors , with proper headings.**

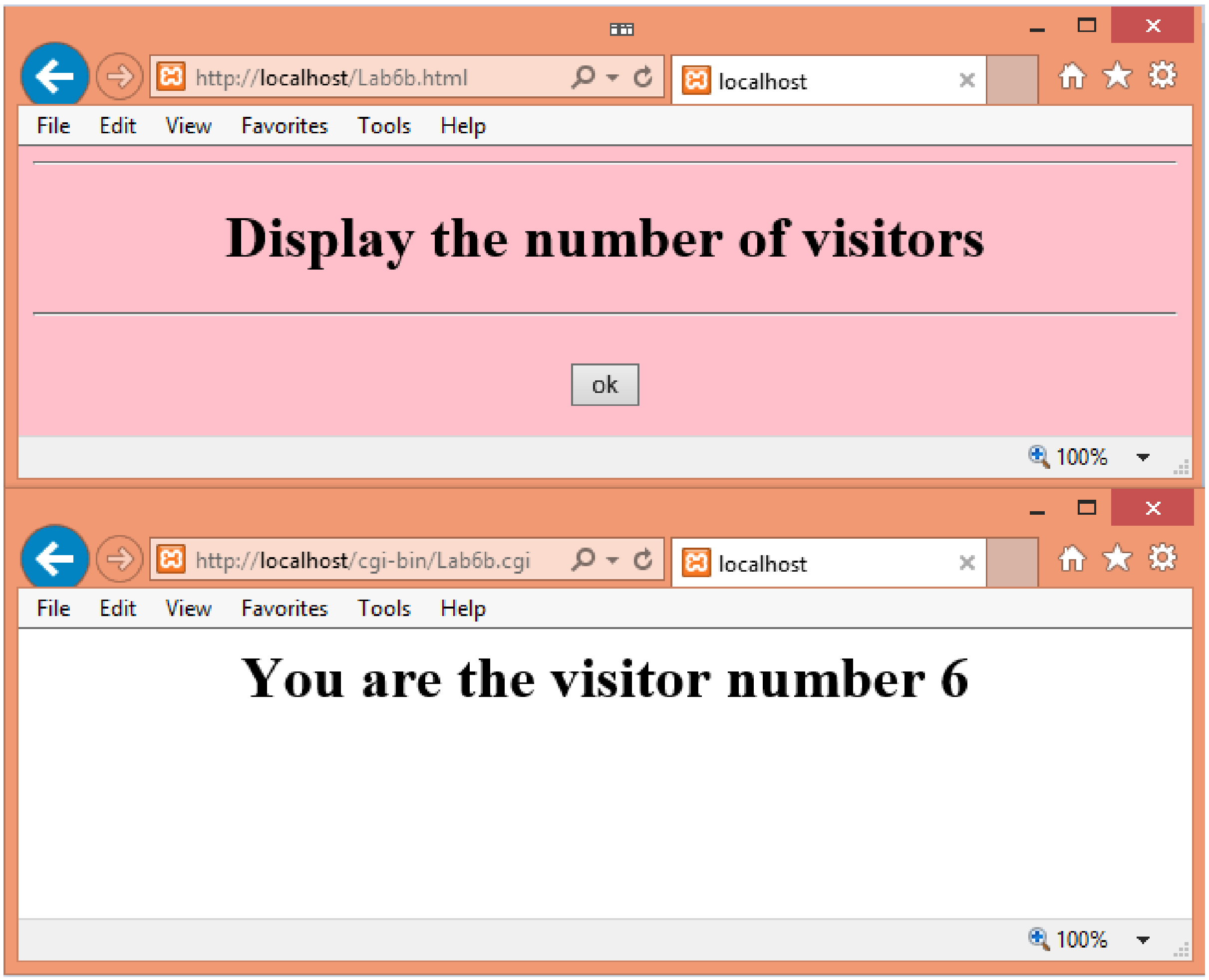
## Perl Script: Lab6b.cgi

```
#!"C:\xampp\perl\bin\perl.exe"

print "content-type: text/html \n\n";

open (FILE, "<visit.txt");

$count=<FILE>;

close(FILE);

open(FILE, ">visit.txt");

$count++;

print FILE "$count";

close(FILE);

print "<center><h1>You are the visitor number $count";
```

## Html : Lab6b.html

```
<html><body bgcolor="pink">

<center><hr><h1>Display the number of visitors</h2><hr>

<form action="/cgi-bin/Lab6b.cgi">

<br><input type="submit" value="OK"><br></center>

</form>

</body>

</html>
```

**OUTPUT: Lab6b.html**


http://localhost/Lab6b.html
localhost
File Edit View Favorites Tools Help
ok
100%
http://localhost/cgi-bin/Lab6b.cgi
localhost
File Edit View Favorites Tools Help
100%

Display the number of visitors

You are the visitor number 6

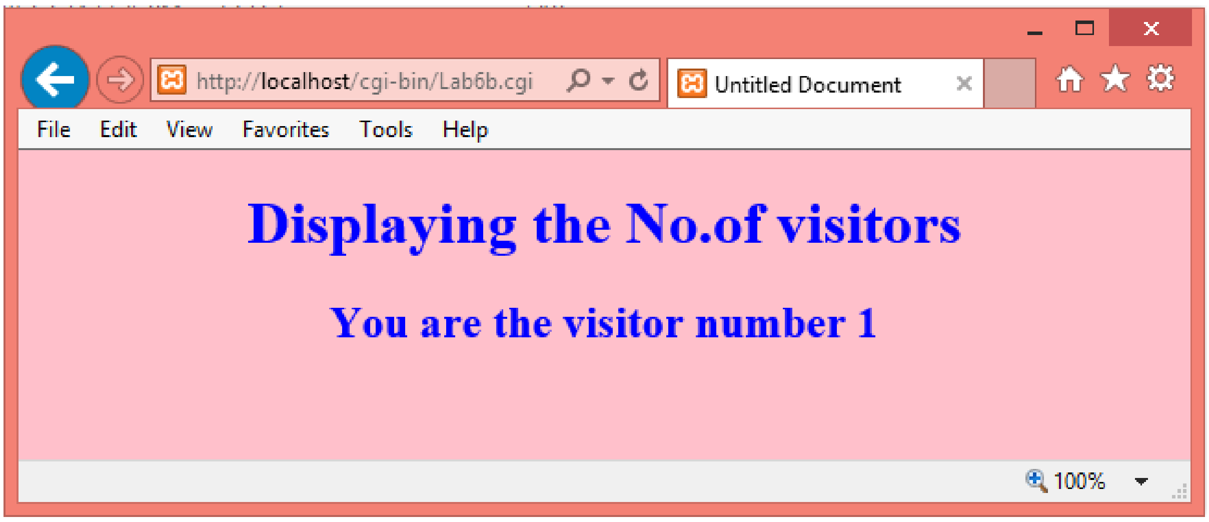
**OR**

**Perl Script: Lab6b.cgi**

```
#!"C:\xampp\perl\bin\perl.exe"

use CGI':standard';

print header();

print start_html(-bgcolor=>"pink", -text =>"blue");

print start_form();

print "<center><h1>Displaying the No.of visitors</h1>";

open(FILE, "<visit.txt");

$count=<FILE>;

close(FILE);
```

```
open(FILE, ">visit.txt");

$count++;

print FILE "$count";

close(FILE);

print "<h2>You are the visitor number $count</h2></center>";

print end_form();

print end_html();
```

## OUTPUT: Lab6b.cgi


http://localhost/cgi-bin/Lab6b.cgi
Untitled Document
File Edit View Favorites Tools Help
100%

Displaying the No.of visitors

You are the visitor number 1

**7) Write a Perl program to display a digital clock which displays the current time of the server.**

In this program a digital clock which displays current time of the server has to be displayed.

- Firstly, the system time is obtained by making use of the localtime ( ) function ($sec,$min,$hr)=localtime(time);

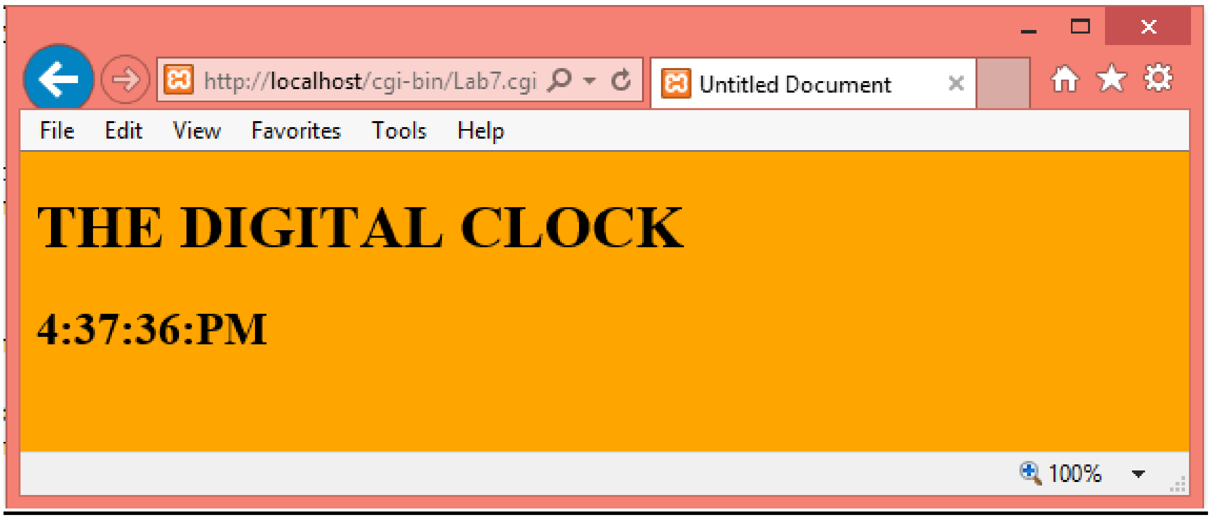
  print “<meta http-equiv='refresh' content='1'>”;
  print "The Current time is:$hr:$min:$sec<br>";
- The concept of Meta tags in HTML is to be known to understand the program. Meta tags are hidden html tags of an html document that are not displayed in a browser but provide a browser or search engine robot with useful information. Meta tags used by the browser define particular things about the document, like what character set to use when displaying the characters on the page, or what language the html document is written it, or how many seconds should be waited before refreshing the page or redirecting to another webpage
- META tags should be placed in the **head** of the HTML document, between the <HEAD> and </HEAD> tags. META tags with an HTTP-EQUIV attribute are equivalent to HTTP headers. Typically, they control the action of browsers, and may be used to refine the information provided by the actual headers. Tags using this form should have an equivalent effect when specified as an HTTP header, and in some servers may be translated to actual HTTP headers automatically or by a pre-processing tool. The meta tag used in the program specifies that the action to be performed is “Refreshing” the page. The content attribute specifies that the pages needs to be refreshed every 1 second.

***NOTE:*** Without using the meta tag, the time is displayed on the page and the “Refresh” button is to be clicked each time to see the updated time. But by making use of the meta tags, the time is updated automatically by the browser. Since the page gets refreshed every second, the “Stop” button in the browser may not be accessible. Hence close the Browser window to stop the program.

## Perl Script: Lab7.cgi

```
#!"C:\xampp\perl\bin\perl.exe"

use strict;

use CGI':standard';

my $ampm;

my($sec,$min,$hour)=localtime();

print header;

print start_html(-bgcolor=>"orange");

print h1("THE DIGITAL CLOCK");

print "<META HTTP-EQUIV='Refresh' CONTENT='1'>";

if($hour>12)

{

     $hour=$hour-12;

     $ampm="PM";

}

else

{

     $ampm="AM";

}

print h2("$hour:$min:$sec:$ampm");

print end_html( );
```

**OUTPUT: Lab7.cgi**


http://localhost/cgi-bin/Lab7.cgi
Untitled Document
File Edit View Favorites Tools Help
100%

THE DIGITAL CLOCK

4:37:36:PM

**8) Write a Perl program to insert name & age information entered by the user into a table created using MySQL & to display the current contents of this table.**

In this program: First create HTML form for accepting the information of User Name and Age information. Once submitted store the values in MySql table and retrieve the table and display the data based on name. Following are the steps :

- It is assumed that a MySQL database and the corresponding tables are ready for use. If not, start the MySQL Server and make everything ready. We have called the database "*employee*" and the table as "*emp*" in this case. The password used for the '*apache* 'user is '*lamp*'.
- DBI (Database Independent Interface) is a Perl Module that provides methods to manipulate SQL Databases. With DBI, one can connect to a database within a Perl script and issue all kinds of queries like select, insert and delete. The "use DBI" pragma makes use of this Perl module.
- The "use strict" pragma makes the user compulsorily declare a variable before it is used. This pragma overrides the Perl's built in feature that states that variables can be used directly without declaration. Therefore with 'strict', if you use a variable before it is declared, an error is generated. Once this pragma is used, the '*my*' keyword is used to declare variables.
- The variable "$dbh" serves as a Database Handler. The connect ( ) function is used to connect to the server. The first parameter indicates that the DBI Perl module is being used to the MySQL database and the database name is 'employee'. The second parameter indicates the user "apache" and the third parameter indicates that the password is "lamp" (Remember that this is the password that we had given with the Grant command. Make sure that this password matches with the one given with the Grant command)
- The variable "$dbh" is the Statement handler that prepares the SQL statement (Query) to be executed.
- The execute ( ) function is used to execute the SQL query.

- The fetchrow ( ) function is used to loop through each record of the table and the records are stored in the variables “$a” and “$b” and these values are printed.
- After this, ensure that you finish the statement handler and disconnect from the server by making use of the finish ( ) and disconnect ( ) functions respectively.

## Html:Lab8.html

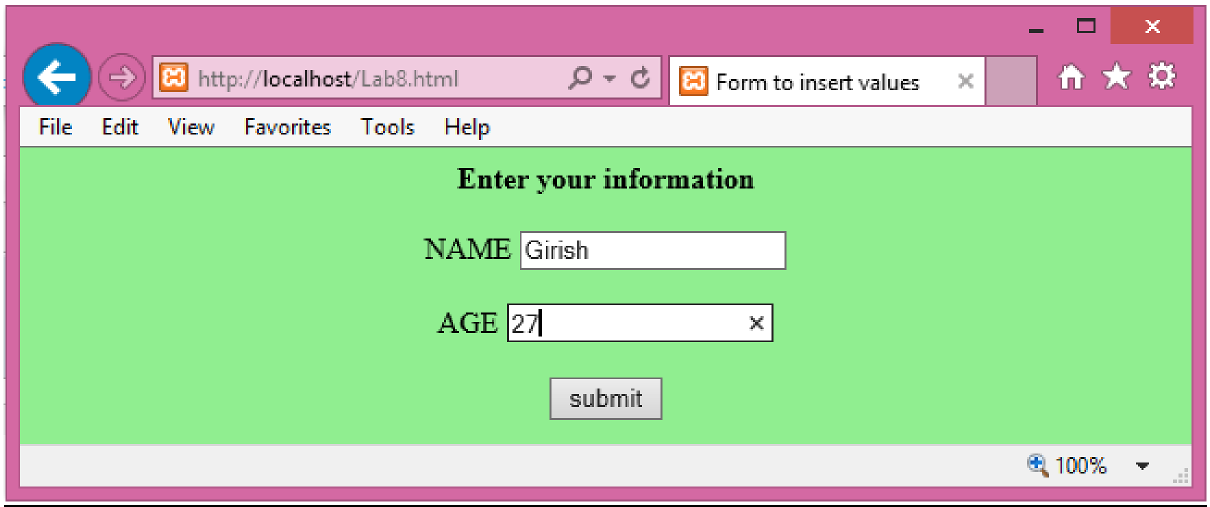
```
<html><body bgcolor="lightgreen">
<form method="get" action="http://localhost/cgi-bin/Lab8.cgi">
<center><b>Enter your information</b><br><br>
NAME <input type="text" name="name"><br><br>
AGE <input type="text" name="age"><br><br>
<input type="submit" value="submit">
</center></form>
</body></html>
```

## Perlscript: Lab8.cgi

```
#!"C:\xampp\perl\bin\perl.exe"
use CGI':standard';
use DBI;
print "Content-type:text/html \n\n";
$nam=param("name");
$age=param("age");
$con=DBI->connect("DBI:mysql:college","root","");
$res=$con->prepare("insert into student values('$nam','$age')");
$res->execute();
$res=$con->prepare("select * from student");
$res->execute();
print "<html><body bgcolor=lightgreen>";
print "<center>THE CONTENTS OF THE DATABASE TABLE IS SHOWN
BELOW<br><br><table border=1>";
print "<tr><th>NAME</th><th>AGE</th></tr>";
while(@a=$res->fetchrow_array())
```

```
{
     print "<tr><td>$a[0]</td><td>$a[1]</td></tr>";
}
print "</table></center>";
print res->finish();
print $con->disconnect();
print "</body></html>";
```

**OUTPUT: Lab8.html**

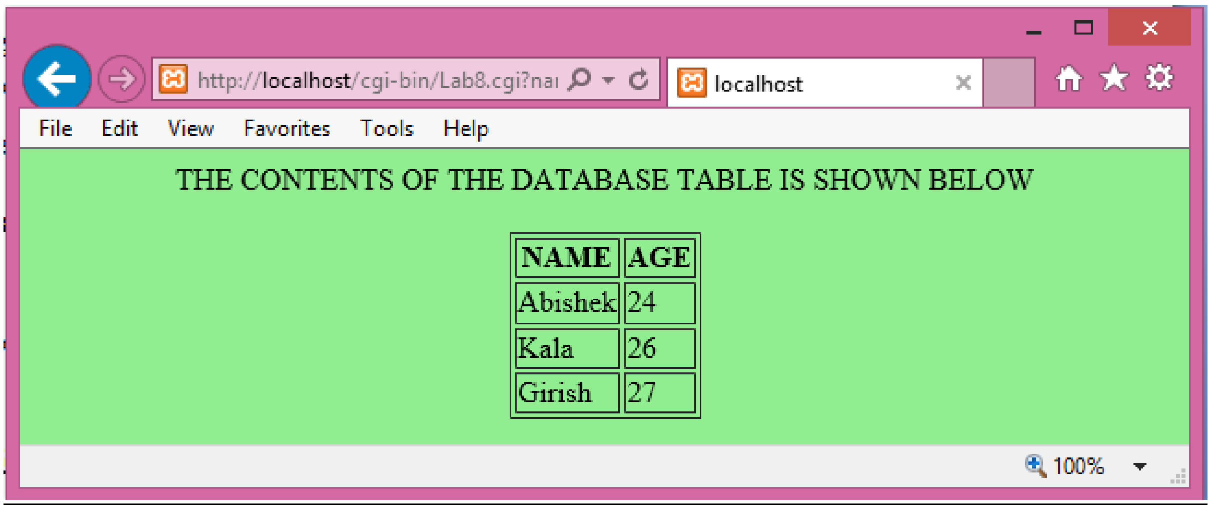

http://localhost/Lab8.html
Form to insert values
File Edit View Favorites Tools Help
100%

**Enter your information**

NAME Girish

AGE 27

submit


http://localhost/cgi-bin/Lab8.cgi?nai
localhost
File Edit View Favorites Tools Help
100%

THE CONTENTS OF THE DATABASE TABLE IS SHOWN BELOW

| NAME | AGE |
|---|---|
| Abishek | 24 |
| Kala | 26 |
| Girish | 27 |

**9) Write a PHP program to store current date-time in a COOKIE and display the Last visited on date-time on the web page upon reopening of the same page**

PHP transparently supports HTTP cookies. Cookies are a mechanism for storing data in the remote browser and thus tracking or identifying return users. Cookies are small text files that are sent by the server to the browser. This is used for session handling which is very important in web development.

They house the client’s state information. We can set cookies using the setcookie() or setrawcookie() function. Cookies are part of the HTTP header, so setcookie() must be called before any output is sent to the browser.

**Syntax:**

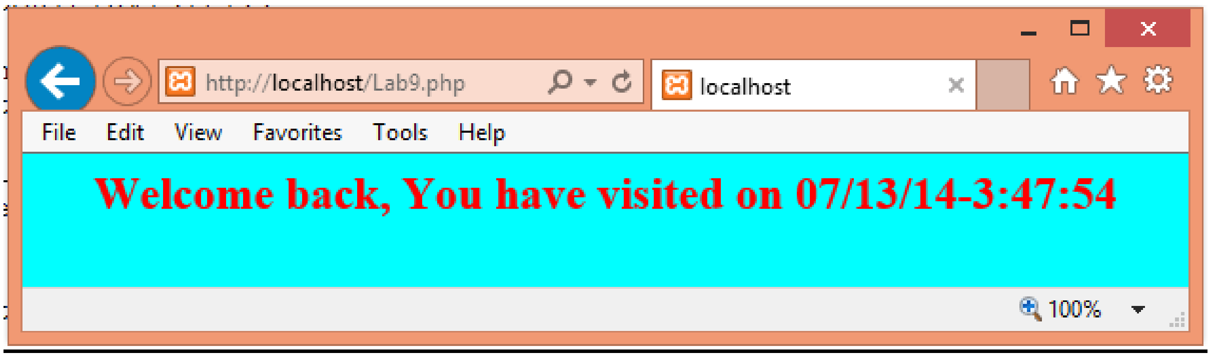
setcookie (name,value,expire,path,domain);

The PHP $_COOKIE variable is used to retrieve a cookie value.

## PHP Script: Lab9.php

```
<html>
<body bgcolor="aqua" text="red">
<?php
$duration=time()+60*60*24*60;
$found=0;
$visit=0;
if(isset($_COOKIE[$visit]))
{
     $found=1;
     $lastvisit=$_COOKIE[$visit];
}
setcookie($visit,date("m/d/y-G:i:s"),$duration);
print "<center>";
if($found==1)
{
     print "<h2>Welcome back, You have visited on
$lastvisit</h2>";
}
else
{
     print "<h3>Welcome to this website</h3>";
}
print "</center>";
?>
</body>
</html>
```

**OUTPUT:Lab9.php**



```
Welcome back, You have visited on 07/13/14-3:47:54
```

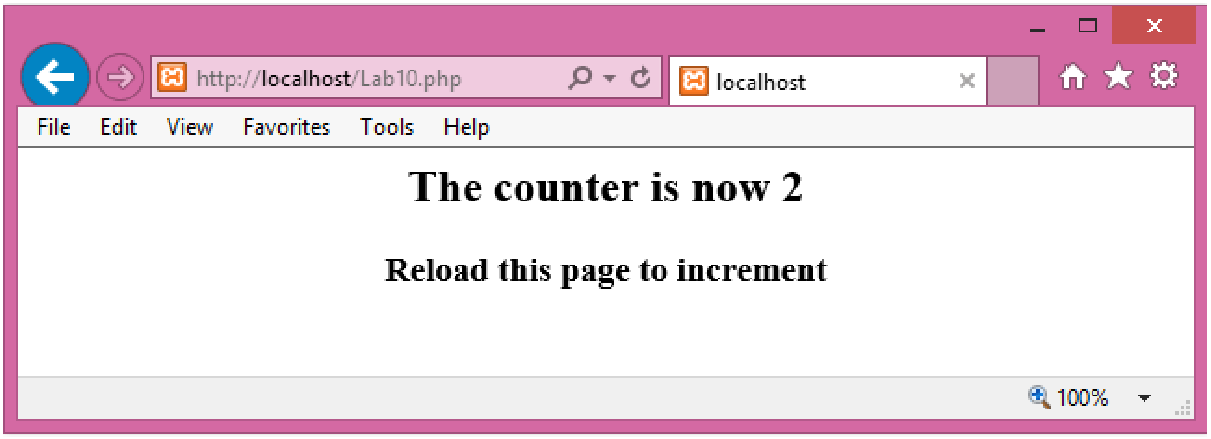
**10) Write a PHP program to store page views count in SESSION, to increment the count on each refresh, and to show the count on web page.**

- A PHP session variable is used to store information about, or change settings for a user session. Session variables hold information about one single user, and are available to all pages in one application.
- A Session provides a mechanism to store the state information between multiple interactions between a Client and a Server. They are used as HTTP is a “Stateless Protocol”, i.e. each request is independent of other.
- Here we try to get the current session first. If there is no session currently, server will automatically create a session when you request.

## PHP Script: Lab10.php

```
<?php

session_start();

print "<center>";

if(!isset($_SESSION))

{

     $_SESSION["count"]=0;

     echo "counter initialized \n";

}

else

{

     $_SESSION["count"]++;

}

echo "The counter is now <b> $_SESSION[count]</b>" . " <p>
Reload this page to increment </p>";

print "</center>";

?>
```

**OUTPUT:Lab10.php**


http://localhost/Lab10.php
localhost
File Edit View Favorites Tools Help
100%

```
The counter is now 2

Reload this page to increment
```

**11) Create an XHTML form with Name, Address Line 1, Address Line 2 and E-mail text fields. On submitting, store the values in MySQL table. Retrieve and display the data based on Name.**

In this program : First create HTML form for accepting the information of Name, Address line1,Address line2 & Email. Once submitted store the values in MySql table and retrieve the table and display the data based on name. Following are the steps :

- connect to the database
- Get the information from webpage
  $accession_no = $_GET [“accession_no”];
  $edition = $_GET [“edition”];
  …
  $publication = $_GET [“publication”];
  If( $accession_no = = “ “ or $edition = = “ “ or ……….)
  { print “Wrong or no input data”;
  Die (“ Record not inserted ”);
- To insert information into the database

## Html:Lab11.html

```
<html>

<body bgcolor="blue">

<h2> Form to insert values</h2>

<form action="Lab11.php" method="post">

Name : <input type="text" name="name" /><br><br>

Add1 : <input type="text" name="addr1" /><br><br>

Add2 : <input type="text" name="addr2" /><br><br>

Email: <input type="text" name="email" /><br><br>

<input type="submit" value="submit" />

<input type="reset" value="reset" />

</form>

</body></html>
```

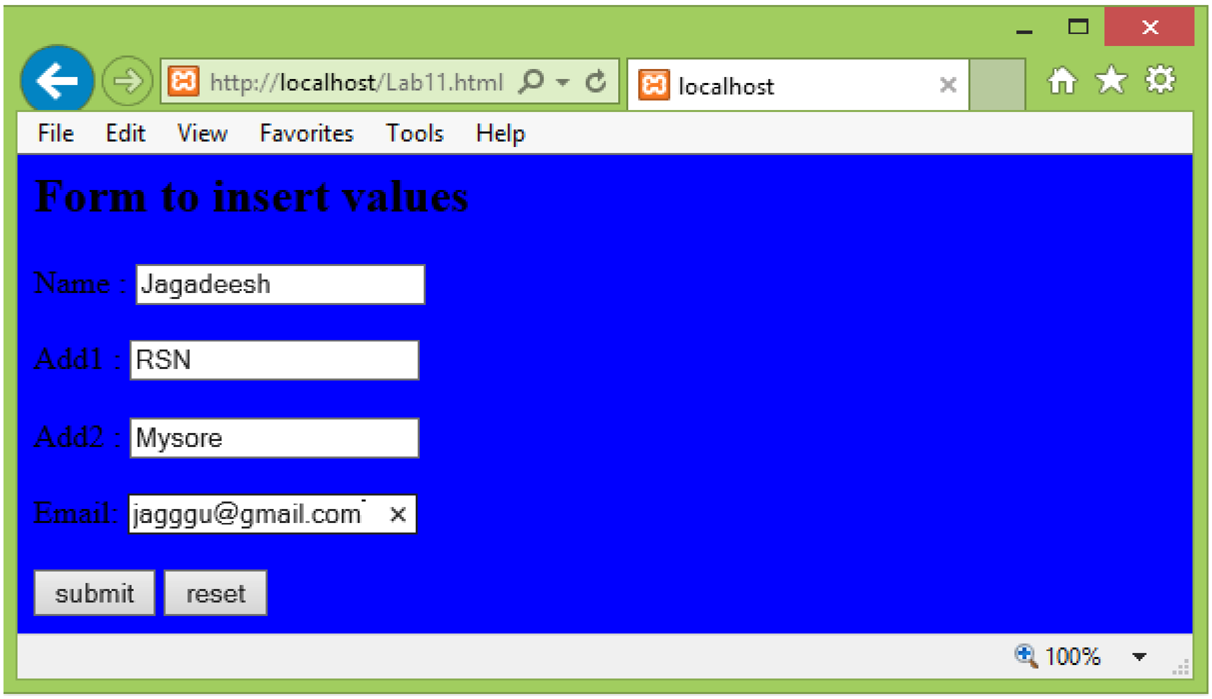
## PHP script:Lab11.php

```
<html>

<body>

<?php

$con = mysql_connect("localhost","root","");

if (!$con)

{

     die('Could not connect: ' . mysql_error());

}

mysql_select_db("college");

$sql="INSERT INTO person (name, addr1, addr2, email)

VALUES ('$_POST[name]','$_POST[addr1]','$_POST[addr2]',
'$_POST[email]')";

if (!mysql_query($sql,$con))

{

     die('Error: ' . mysql_error());

}

echo "1 record added";

mysql_close($con)

?>

<h2> Form to retrieve values based on name</h2>

<form action="Lab11a.php" method="post">

Name: <input type="text" name="name" />

<input type="submit" />

</form>

</body>

</html>
```

## PHP script:Lab11a.php

```
<html><body><?php

$con = mysql_connect("localhost","root","");

if (!$con)

{

     die('Could not connect: ' . mysql_error());

}

mysql_select_db("college");

$result = mysql_query("SELECT * FROM person where
name='$_POST[name]'");

if(!$result)

{

     echo "There is no records";

}

echo "<table border='1'>

<tr><th>Name</th><th>Addr1</th><th>Addr2</th><th>Email</th></tr>";

while($row = mysql_fetch_array($result))

{

     echo "<tr>";

     echo "<td>" . $row['name'] . "</td>";

     echo "<td>" . $row['addr1'] . "</td>";

     echo "<td>" . $row['addr2'] . "</td>";

     echo "<td>" . $row['email'] . "</td>";

     echo "</tr>";

}

echo "</table>";

mysql_close($con);

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

**OUTPUT:Lab11.html**

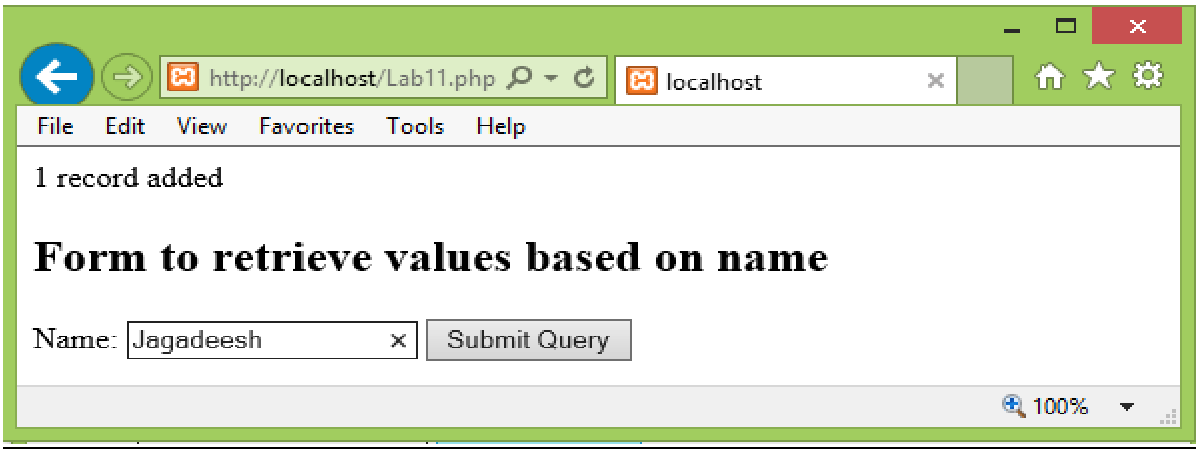

http://localhost/Lab11.html
localhost
File Edit View Favorites Tools Help
100%

Form to insert values

| | |
|---|---|
| Name : | Jagadeesh |
| Add1 : | RSN |
| Add2 : | Mysore |
| Email: | jagggu@gmail.com |

submit reset

**Lab11.php**

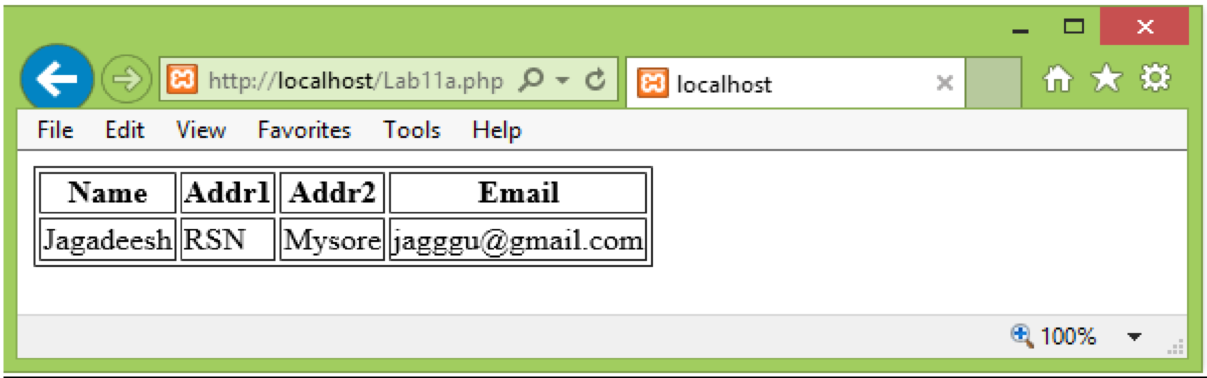

http://localhost/Lab11.php
localhost
File Edit View Favorites Tools Help
100%

1 record added

Form to retrieve values based on name

Name: Jagadeesh Submit Query

**Lab11a.php**


http://localhost/Lab11a.php
localhost
File Edit View Favorites Tools Help
100%

| Name | Addr1 | Addr2 | Email |
|---|---|---|---|
| Jagadeesh | RSN | Mysore | jagggu@gmail.com |

**12)Build a Rails application to accept book information viz. Accession number, title, authors, edition and publisher from a web page and store the information in a database and to search for a book with the title specified by the user and to display the search results with proper headings.**

**Steps to be followed:**

**Step 1:**Open Ruby console window from Instant Rails. In the command prompt use the below code and create databases/tables and quit from mysql.

mysql -u root

**create database lab12_development;**

**use lab12_development;**

**create table books** (id int not null auto_increment, accno int not null, title varchar(30) not null, author varchar(30) not null, edition int not null, publisher varchar(30) not null,primary key(id));

After creating the database and table exit from mysql,by typing the command **exit**

**Step 2:** In the command prompt type the below command to create a new project named lab12.

**rails -d mysql lab12** //database name and this should be same **Step 3:**Goto the **lab12** folder by typing the command **cd lab12**.
**Step 4:**Type the below command to create the scaffold.

**Note:** Name in scaffold command should be singular of table name with first letter caps.

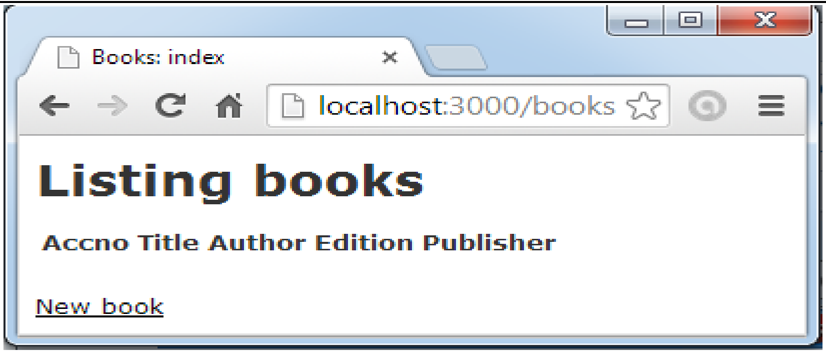
**ruby script/generate scaffold Book accno:int title:string author:string edition:int publisher:string**

**Step 5:** Start the WEBrik server by typing the command.

**ruby script/server** **Step 6:** Open the IE browser and type the below address to run the project.

**http://localhost:3000/books**

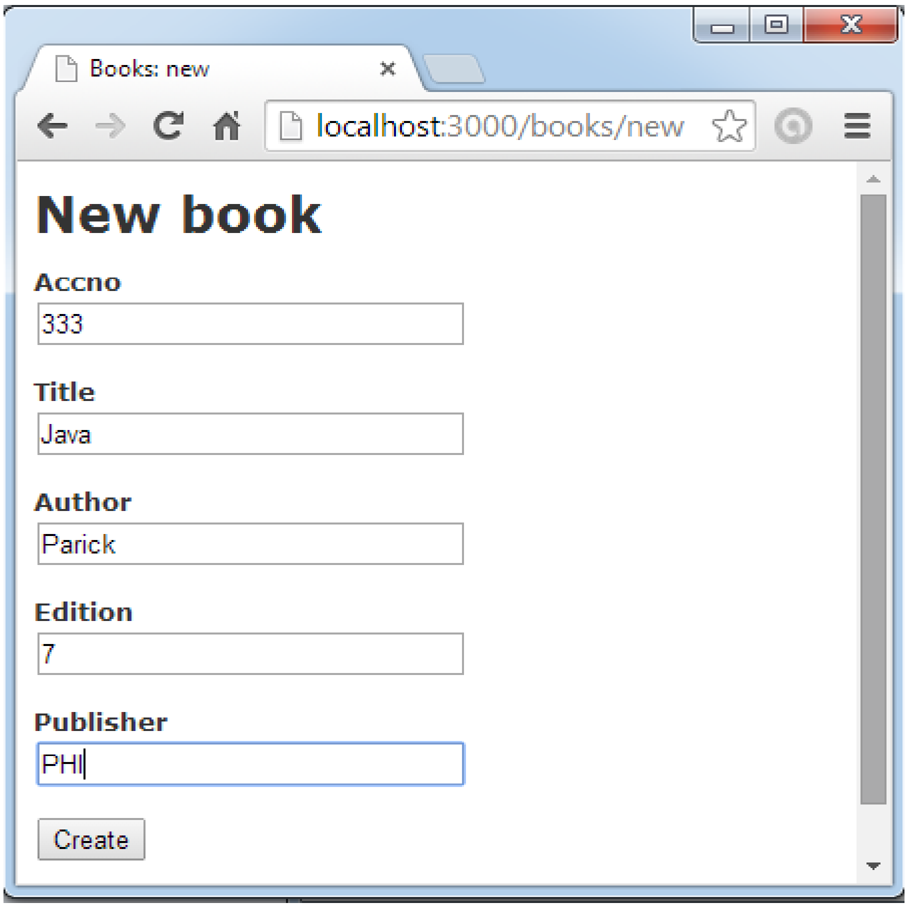
A page opens in the browser. Create new entries of books by clicking **New book**. You can insert, edit, and delete any no. of books here.



Listing books

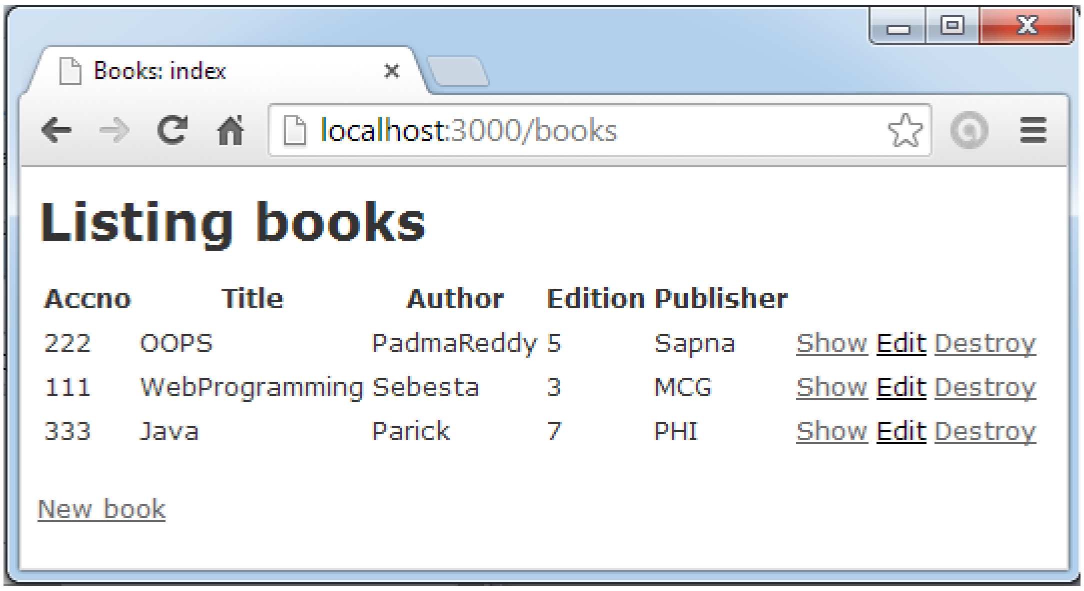
Accno Title Author Edition Publisher

New book



New book

| | |
|---|---|
| Accno | 333 |
| Title | Java |
| Author | Parick |
| Edition | 7 |
| Publisher | PHI |



Listing books

| Accno | Title | Author | Edition | Publisher | | | |
|---|---|---|---|---|---|---|---|
| 222 | OOPS | PadmaReddy | 5 | Sapna | Show | Edit | Destroy |
| 111 | WebProgramming | Sebesta | 3 | MCG | Show | Edit | Destroy |
| 333 | Java | Parick | 7 | PHI | Show | Edit | Destroy |

New book

**Step 7:** Open one more Ruby console window using Instant Rails and navigate to the **lab12** folder.

**Step 8:** Inside the **lab12** folder type the below command for creating a controller named main.

**ruby script/generate controller main**

**Step 9:** Open the file **main_controller.rb** from **C:\InstantRails-2.0-win\rails_apps\lab12\app\controllers** folder and replace the contents with the below code.

```
class MainController < ApplicationController

  def welcome

     @num_books = Book.count

  end



  def result

     @booktitle = params[:stitle]

     @bookz   =   Book.find(:all,   :conditions   =>   "title   =
     '#{@bookid}'")

  end

end
```

**Step 10:** Goto the folder **C:\InstantRails-2.0-win\rails_apps\lab12\ app\views\main** and create a file named result.rhtml and paste the below code for creating a view.

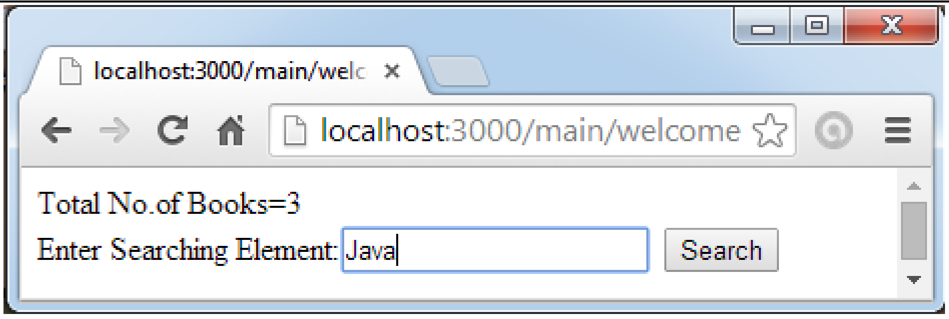
```
<html>

<title> Welcome template for books </title>

<body>

<p> Entered book title is <%= @booktitle %> </p>

<table border=1>

<tr><th>BookId</th><th>AccNo</th><th>Title</th><th>Author</th>
<th>Edition</th><th>Publisher</th></tr>

  <% @bookz.each do |bk|

     @id = bk.id

     @accno=bk.accno
     @title=bk.title
     @edition=bk.edition
     @publisher=bk.publisher  %>
     <tr> <td> <%= @id %></td>
          <td><%= @accno %> </td>
          <td><%= @title %></td>
          <td> <%= @author %></td>
          <td> <%= @edition %></td>
          <td> <%= @publisher %></td>
       </tr>
     <% end %>
</table>
</body>
</html>
```

**Step 11:** Goto the folder **C:\InstantRails-2.0-win\rails_apps\ lab12\app\ views\main** and create a file named <u>welcome.rhtml</u> and paste the below code for creating a view to display the results.

```
<html>
<title> Welcome template for books </title>
<body><p> Total number of books = <%= @num_books %> </p>
<form action = "result" >
Enter Searching Element: <input type="text" name="stitle" />
<input type=submit value="Search" />
</form>
</body>
</html>
```

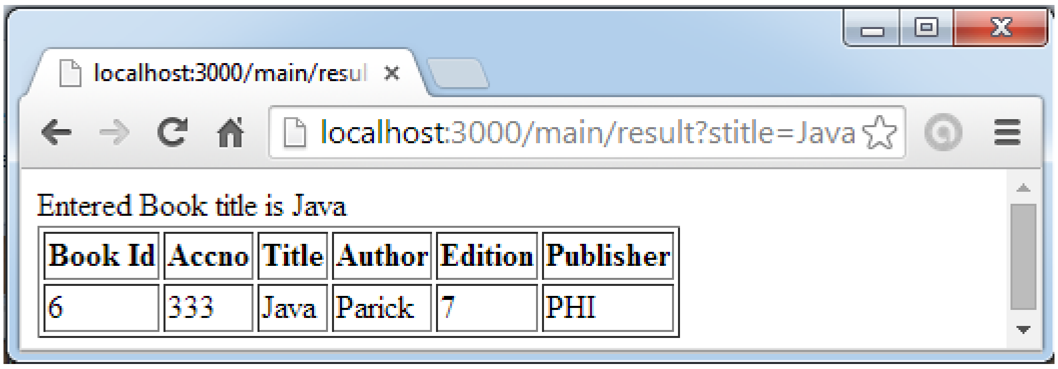
**Step 12:** To search the result go to browser and type the following

http://localhost:3000/main/welcome



Total No.of Books=3

Enter Searching Element: Java [Search]


localhost:3000/main/resul
localhost:3000/main/result?stitle=Java

Entered Book title is Java

| Book Id | Accno | Title | Author | Edition | Publisher |
|---|---|---|---|---|---|
| 6 | 333 | Java | Parick | 7 | PHI |

## VIVA QUESTIONS

1. Why HTML is called a “markup” language?

2. Can you name some Markup languages other than HTML?

3. How do you write comments in HTML?

4. What does the <br> and the <p> tag in HTML do?

5. How do you insert bullets in a HTML page?

6. What is the difference between Dynamic HTML and Dynamic Web Pages?

7. What are the Components of URL?

8. What do you mean by the “Shebang Line”?

9. What does ‘80’ mean in localhost: 8080?

10. Difference between print “$h” and print ‘$h’?

11. How do you declare arrays in Perl? Comment on how the print ( ) works with arrays.

12. What does the “< = >” operator do in Perl?

13. What does the keyword “strict” mean?

14. How do you execute a UNIX command in Perl?

15. What is the difference between system( ) and backticks(`)?

16. Why the “\n\n” is used in print “Content-type:text/html\n\n”;? Will the program execute even without it?

17. What does the IP address 127.0.0.1 correspond to?

18. What is DBI? What is “independent” about it?

19. What does “standard” do? What is the alternative to “standard”?

20. Differentiate between Servlets and CGI.

21. What does the PrintWriter ( ) method do in ServletProgramming?

22. What is the difference between “GenericServlet” and the “HttpServlet” classes?

23. What are Applets?

24. What are cookies? Why do you need them?

25. What do you mean by a “Session”? Comment on its importance.

26. What do you mean by URI? Is it the same as URL?

27. What is a Query String?

28. What do you mean by “Embedded HTML”?

29. What does the “-w” option in the Interpreter line in Perl Programs do?

30. How do you declare functions in Perl?

31. How do you write comments in Perl?

32. What does the “short-circuit” logic operator used in languages like Perl, Java do?

33. Why do you have to “use” a database that you have created in MySQL before you perform other operations?

34. What are the different data types supported by MySQL?

35.What does the “grant” command in MySQL do?

36. What does the die ( ) do?

37. What are the benefits of using Perl?

38. What are the advantages of using CGI?

39. Why is that the CGI Programs are also stored with a “.pl” extension? Can the extension be changed to “.cgi”?

40. What does the chomp ( ) do?

41. What do you mean by the “here” document? Comment on its uses.

42. What do you mean by the “Path Information”?

43. What is the difference between “GET” and “POST”?

44. What is the difference between doGet ( ) and doPost ( ) methods used in Servlet Programming?

45. What does the param ( ) function do?

46.What do you mean by SSI? What are its uses?

47. Why is Java said to be “secure”?

48. What do you mean by “Exception Handling” in Java?

**Viva Questions with answers:**

**1. HTML stands for**

As: Hyper Text Markup Language

**2. What type of language is HTML?**

As: Markup Language

**3. What does an HTML document describe?**

As: Web pages

**4. Links in HTML are defined with what tag?**

As: <a>, anchor tag

**5. How many heading styles are there in HTML?**

As: 6 headings, h1..h6

**6. How line break is given in HTML?**

As: using <br> tag

**7. What is the use of attributes used in HTML?**

As: They provide additional information for tags

**8. For what purpose <hr> tag is used?**

As: The <hr /> tag is used to create an horizontal rule (line).

**9. What are the basic text formatting tags used in HTML?**

As: <b> for bold, <i> for italic, <u> for underline etc.

**10. What is a hyperlink?**

As: In web terms, a hyperlink is a reference (an address) to a resource on the web.

**11. How images are displayed on web pages in HTML?**

As: In HTML, images are defined with the <img> tag.

The <img> tag is empty, which means that it contains attributes only and it has no closing tag.

To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display on your page. The syntax of defining an image: <img src=”url/”>

**12. Explain <table> tag used in HTML?**

As: Tables are defined with the <table> tag. A table is divided into rows (with the <tr> tag), and each row is divided into data cells (with the <td> tag). The letters td stands for "table data," which is the content of a data cell. A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.

**13. How many types of lists are available in HTML?**

As: three types of lists are available <ul> for unordered list, <ol> for ordered list, <dl> for definition lists.

**14. Explain <form> element used in HTML?**

As: A form is an area that can contain form elements.

Form elements are elements that allow the user to enter information (like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.) in a form.A form is defined with the <form> tag.

**15. How colors are defined in HTML?**

As: HTML colors are defined using a hexadecimal (hex) notation for the combination of Red, Green, and Blue color values (RGB). The lowest value that can be given to one of the light sources is 0 (hex 00). The highest value is 255 (hex FF). Hex values are written as 3 double digit numbers, starting with a # sign.

**16. Explain about frames used in HTML?**

As: With frames, you can display more than one HTML document in the same browser window. Each HTML document is called a frame, and each frame is independent of the others.The disadvantages of using frames are:

- The web developer must keep track of more HTML documents
- It is difficult to print the entire page

**17. How to use styles in HTML?**

As: When a browser reads a style sheet, it will format the document according to it. There are three ways of inserting a style sheet:

External Style Sheet

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing one file. Each page must link to the style sheet using the <link> tag. The <link> tag goes inside the head section.

Internal Style Sheet

An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section with the <style> tag.

**18. Expand CSS.**

As: Cascading Style Sheets

**19. What does styles do in Style sheets ?**

As: They tell how to display various elements

**20. How CSS syntax is made up of?**

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

**21. What is Javascript?**

As: JavaScript is *THE* scripting language of the Web.JavaScript is used in millions of Web pages to add functionality, validate forms, detect browsers, and much more.

**22. What Javascript can do?**

As: JavaScript gives HTML designers a programming tool

JavaScript can put dynamic text into an HTML page

JavaScript can react to events

JavaScript can read and write HTML elements –

JavaScript can be used to validate data –

JavaScript can be used to detect the visitor's browser –

JavaScript can be used to create cookies

**23. What was the original name of Javascript?**

As: ECMA Script (European Computers Manufacturers Association)

**24. What is a Javascript statement?**

As: A JavaScript statement is a command to a browser. The purpose of the command is to tell the browser what to do.

**25. Explain the importance of XML?**

As: XML stands for EXtensible Markup Language.

XML was designed to transport and store data.

XML is important to know, and very easy to learn.

XML tags are not predefined, user has to define the tags

**26. What is the difference between XML and HTML?**

As: XML is not a replacement for HTML.

XML and HTML were designed with different goals:

- XML was designed to transport and store data, with focus on what data is.
- HTML was designed to display data, with focus on how data looks.

HTML is about displaying information, while XML is about carrying information.

**27. What is XML tree?**

As: XML Documents Form a Tree Structure

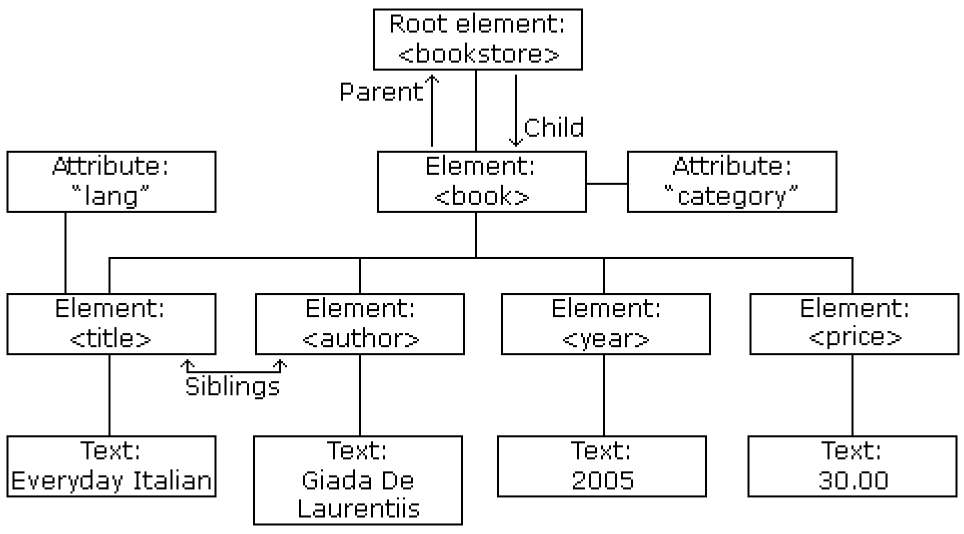
XML documents must contain a root element. This element is "the parent" of all other elements.

The elements in an XML document form a document tree. The tree starts at the root and branches to the lowest level of the tree.

All elements can have sub elements (child elements):

The terms parent, child, and sibling are used to describe the relationships between elements. Parent elements have children. Children on the same level are called siblings (brothers or sisters).

All elements can have text content and attributes (just like in HTML).



**28. What are the syntax rules for XML document?**

As: All XML Elements Must Have a Closing Tag

XML Tags are Case Sensitive

XML Elements Must be Properly Nested

XML Documents Must Have a Root Element

**29. What is an XML Element?**

As: An XML element is everything from (including) the element's start tag to (including) the element's end tag.

An element can contain other elements, simple text or a mixture of both. Elements can also have attributes.

```
<bookstore><book
category="CHILDREN">
<title>Harry Potter</title>

 <author>J
 Rowling</author>
 <year>2005</year>
 <price>29.99</price>
 </book>                <bo
 category="WEB">
 <title>Learning  XML</titl
 <author>Erik
 Ray</author>
 <year>2003</year>
 <price>39.95</price>
 </book> </bookstore>
```

In the example above, <bookstore> and <book> have **element contents**, because they contain other elements. <author> has **text content** because it contains text.

In the example above only <book> has an **attribute** (category="CHILDREN").

**30. What is a well formed XML document?**

**As:** Well Formed XML Documents

A "Well Formed" XML document has correct XML syntax.

- XML documents must have a root element
- XML elements must have a closing tag
- XML tags are case sensitive

 XML elements must be properly nested

 XML attribute values must be quoted

**31. What is a valid XML document?**

As: A "Valid" XML document is a "Well Formed" XML document, which also conforms to the rules of a Document Type Definition (DTD):

<?xml version="1.0" encoding="ISO-
8859-1"?> <!DOCTYPE note SYSTEM
"Note.dtd"> <note> <to>Tove</to>
<from>Jani</from>
<heading>Reminder</heading>
<body>Don't forget me this
weekend!</body> </note>

The DOCTYPE declaration in the example above, is a reference to an external DTD file. The content of the file is shown in the paragraph below.

**32. What is the purpose of XML DTD?**

As: The purpose of a DTD is to define the structure of an XML document. It defines the structure with a list of legal elements:

**33. What is PHP?**

As: PHP is a powerful tool for making dynamic and interactive Web pages.

PHP is the widely-used, free, and efficient alternative to competitors such as Microsoft's ASP.

 PHP stands for**P**HP: **H**ypertext **P**reprocessor

 PHP is a server-side scripting language, like ASP

 PHP scripts are executed on the server

 PHP supports many databases (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.)

 PHP is an open source software

 PHP is free to download and use

**34. Why PHP is popular than ASP?**

As: PHP runs on different platforms (Windows, Linux, Unix, etc.)

PHP is compatible with almost all servers used today (Apache, IIS, etc.)

PHP is FREE to download from the official PHP resource: www.php.net

PHP is easy to learn and runs efficiently on the server side

**35. What is basic PHP syntax?**

As: A PHP scripting block always starts with **<?php** and ends with **?>**. A PHP scripting block can be placed anywhere in the document.On servers with shorthand support enabled you can start a scripting block with <? and end with ?>.For maximum compatibility, we recommend that you use the standard form (<?php) rather than the shorthand form.

**36. Expand Perl?**

As: **Perl (Practical Extraction and Reporting Language)** A scripting language for web servers. Most often used on Unix servers

**37. What is CGI script?**

As: CGI scripts are executables that will execute on the server to produce dynamic and interactive web pages. Most ISPs offer some kind of CGI capabilities. ISPs often offer preinstalled, ready to run, guest-books, page-counters, and chat-forums solutions in CGI.CGI is most common on Unix or Linux servers.

**38. Who developed PHP?**

As: Rasmus Lerdorf

**39. Who developed HTML and World Wide Web?**

As: Tim Berners Lee

**40. What is the use of HTTP protocol?**

As: application-level protocol for distributed, collaborative, hypermedia information systems

- generic, stateless, object-oriented
- can be used for many tasks, such as name servers & distributed object management systems