LIST OF PROGRAMS

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 100 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.

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

SOLUTION:

```
<?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">
<head><title> Fibonacci</title></head>
<body bgcolor="lightgreen">
<script type="text/JavaScript">
var fib1=0,fib2=1,fib=0;
var num = prompt("Enter a number","");
if(num!=null && num>0)
{
        document.write("<h1> The Fibonacci Numbers are </h1><br />");
         if(num==1)
                document. write("<h1> "+ fib1 + "</h1>");
         else
                document.write("<h1>" + fib1 + "</h1>");
                document.write("<h1>" + fib2 + "</h1>");

         for(i=3;i<=num; i++)
        {
                fib= fib1 + fib2;
                document.write("<h1> " + fib + "</h1>");
                fib1=fib2;
                fib2=fib;
        }
}
else
        alert("No Proper Input");
</script>
</body>
</html>
```

EXECUTION PROCEDURE:

1. Login into LINUX OS through root credentials
2. Start Apache web Server

   [root@localhost ~]# systemctl start httpd.service
3. Change the directory to /var/www/html

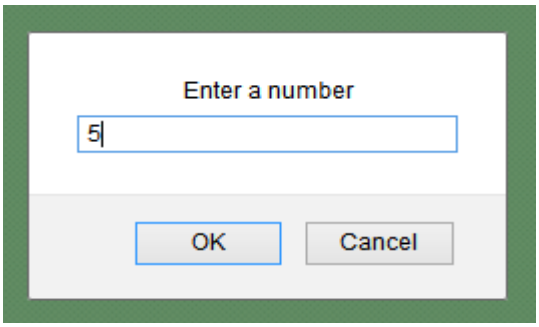
   [root@localhost ~]# cd /var/www/html
4. Type the above XHTML code and save the file.

   [root@localhost html]# gedit Labprogram1a.html
5. Go to web browser (Mozilla Firefox or Konqueror) and type in the URL as

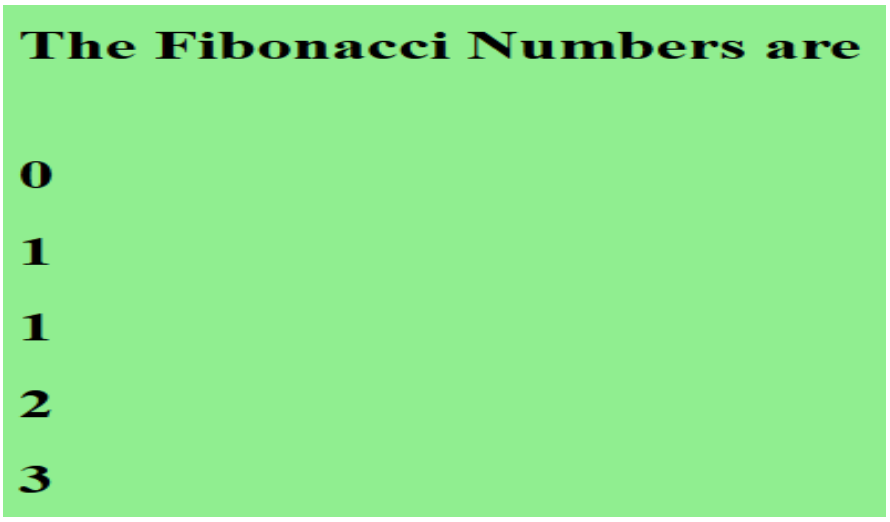
   http://localhost/ Labprogram1a.html [Enter]

OUTPUT:

Screen 1: Enter the 'N' number using prompt



Screen 2: Display the '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.

SOLUTION:

```
<?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">
<head><title>Table of Numbers</title>
</head>
<body bgcolor="lightgreen">
<script type="text/javascript">
var num = prompt("Enter a number", "");
if(num >0 && num !=null)
{
        squareof="Number and its Squares are \n";
        for(i=1;i <= num; i++)
        {
                squareof = squareof +  i  + " = " + i*i + "\n";
        }
        alert(squareof);
}
else
        alert("No input supplied");
</script>
</body>
</html>
```

EXECUTION PROCEDURE:

1. Login into LINUX OS through root credentials
2. Start Apache web Server

   [root@localhost ~]# systemctl start httpd.service
3. Change the directory to /var/www/html

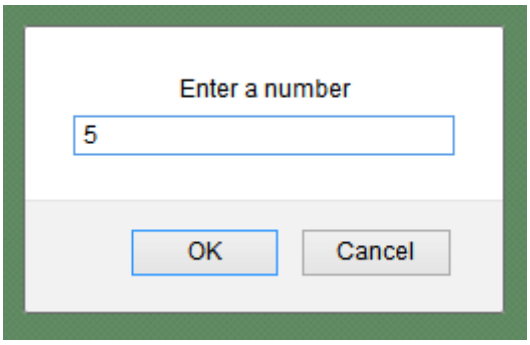
   [root@localhost ~]# cd /var/www/html
4. Type the above XHTML code and save the file.

   [root@localhost html]# gedit Labprogram1b.html

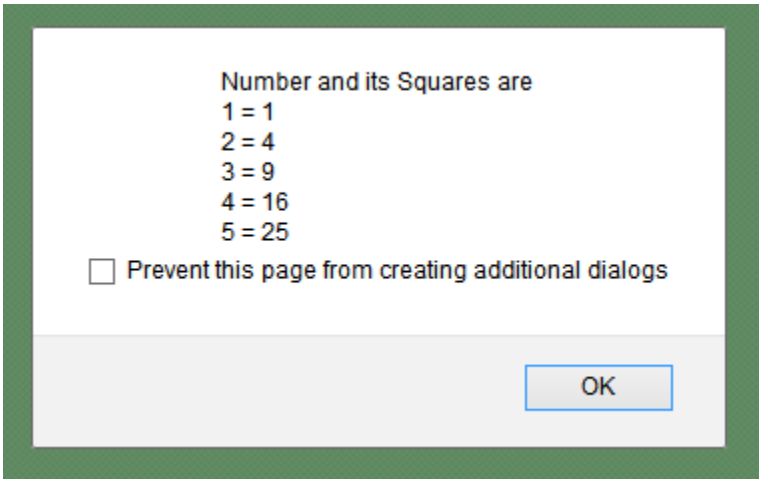
5. Go to web browser (Mozilla Firefox or Konqueror) and type in the URL as
   http://localhost/ Labprogram1b.html [Enter]

OUTPUT:

Screen 1: Enter the 'N' number using prompt.



Screen 2: Displays the tables of numbers with squares using alert.



2. a) Develop and demonstrate, using JavaScript, 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.

SOLUTION:

```
<?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">
<head><title>USN Validation</title>
<script type="text/javascript">
function validateUSN()
{
        var usn = document.getElementById("usn");
        var usnExp=/[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 feild is empty");
                usn.focus();
                return false;
        }
        else if(!usn.value.match(usnExp))
        {
                alert("US Number is not correct");
                usn.focus();
                return false;
        }
        alert("US Number "+usn.value+" IS CORRECT");
        return true;
}
</script>
</head>
<body bgcolor="lightgreen">
<form>
<br/><br/><center>
```

```
<b>Enter USN  </b> <input type="text" id="usn"/> <br /><br />
<input type="submit" value="Check USN" onClick="validateUSN()"/>
</center></form></body></html>
```

EXECUTION PROCEDURE:

1. Login into LINUX OS through root credentials
2. Start Apache web Server

   [root@localhost ~]# systemctl start httpd.service
3. Change the directory to /var/www/html

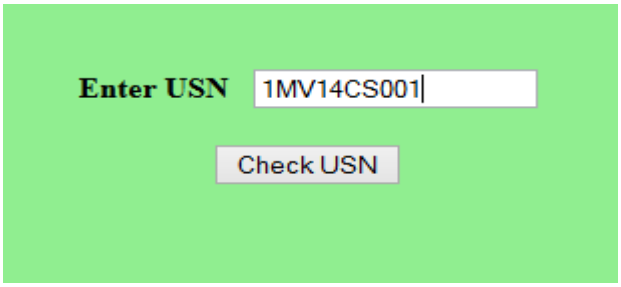
   [root@localhost ~]# cd /var/www/html
4. Type the above XHTML code and save the file.

   [root@localhost html]# gedit Labprogram2a.html
5. Go to web browser (Mozilla Firefox or Konqueror) and type in the URL as

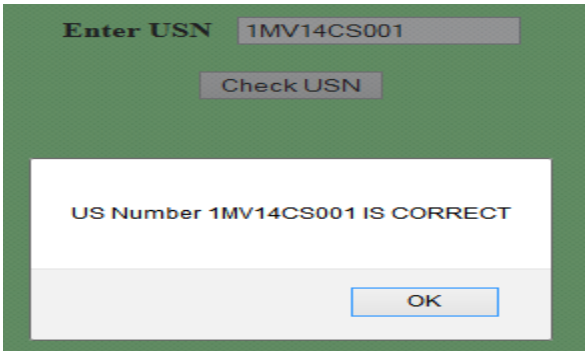
   http://localhost/ Labprogram2a.html [Enter]

OUTPUT:

Screen 1: The web page to enter USN.



Screen 2: USN validation message using alert.



b) Modify the above program to get the current semester also (restricted to be a number from 1 to 8).

SOLUTION:

```
<?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">
<head><title>USN & Sem Validation </title>
<script type="text/javascript">
function validateUSN()
{
        var usn = document.getElementById("usn");
        var sem = document.getElementById("sem");
        var usnExp = /[1-4][A-Z][A-Z][0-9][0-9][A-Z][A-Z][0-9][0-9][0-9]$/;
        var semExp = /[1-8]$/;

        if(usn.value.length == 0 || sem.value.length == 0)
        {
                alert("Please Enter Empty feilds");
                return false;
        }
        else if(!usn.value.match(usnExp))
        {
                alert("Entered US Number is not correct");
                usn.focus();
                return false;
        }
        else if(!sem.value.match(semExp))
        {
                alert("Entered Sem is not correct");
                sem.focus();
                return false;
        }
        alert(" Entered USN "+usn.value+ " & Sem "+sem.value+ " is Correct");
        return true;
}
</script>
</head>
```

```
<body bgcolor="lightgreen">
<form>
<br/><center><b>Enter the following <br/><br/>
USN   <input type="text" id="usn"/> <br /><br />
SEM   <input type="text" id="sem"/></b> <br /><br />
<input type="submit" value="Check USN & SEM" onClick="validateUSN()"/>
</center></form>
</body></html>
```

EXECUTION PROCEDURE:

1. Login into LINUX OS through root credentials
2. Start Apache web Server

   [root@localhost ~]# systemctl start httpd.service
3. Change the directory to /var/www/html

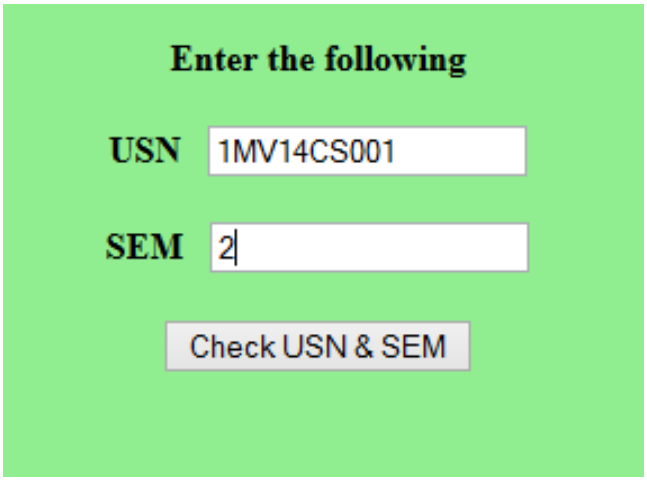
   [root@localhost ~]# cd /var/www/html
4. Type the above XHTML code and save the file.

   [root@localhost html]# gedit Labprogram2b.html
5. Go to web browser (Mozilla Firefox or Konqueror) and type in the URL as

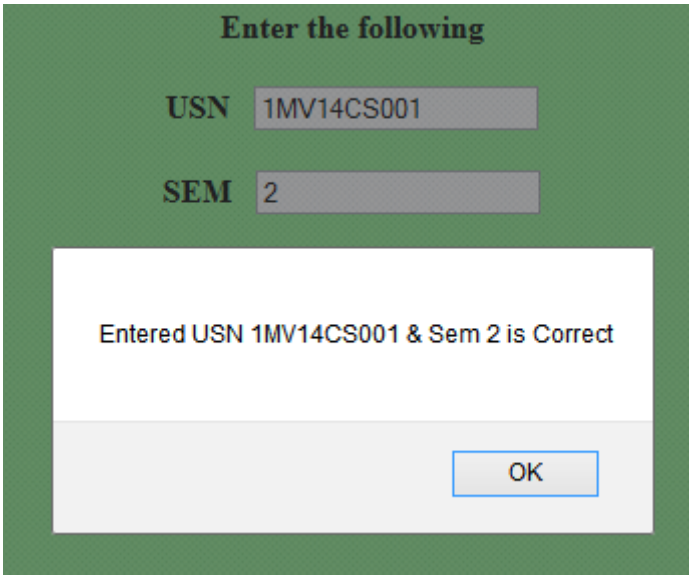
   http://localhost/ Labprogram2b.html [Enter]

OUTPUT:

Screen 1: The web page to enter USN & SEM feilds.



Screen 2: The USN and SEM validation message using alert.



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.

SOLUTION:

```
<?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">
<head><title>Paragraphs stacked on each other</title>
<style type="text/css">
  .para1{width:300px;background-color:green;position:absolute;
        top:100px;left:200px;z-index:0;}
  .para2{width:300px;background-color:cyan;position:absolute;
      top:120px;left:220px;z-index:0;}
  .para3{width:300px;background-color:purple;position:absolute;
      top:140px;left:240px;z-index:0;}
 </style>
 <script type="text/javascript">
 var topLayer="1";
 function mover(toTop)
 {
            var oldTop=document.getElementById(topLayer).style;
             var newTop=document.getElementById(toTop).style;
             oldTop.zIndex="0";
             newTop.zIndex="10";
             topLayer=toTop;
  }
 </script>
 </head>
 <body><body bgcolor="lightgreen">
 <center><h2>Paragraphs stacked on each other shown below</h2>
<div  class="para1" id="1"  onmouseover="mover('1')">
   Rails is a development framework for Web-based applications that access databases. A
   framework is a system in which much of the more-or-less standard software
   parts are furnished by the framework, so they need not be written by the applications
```

```
  developer.
 </div>
 <div class="para2" id="2"   onmouseover="mover('2')">
   The original goal of JavaScript was to provide programming capability at both the server and
   the client ends of a Web connection. Since then, JavaScript has grown into a full-fledged
   programming language that can be used in a variety of application areas.
 </div>
 <div class="para3" id="3" onmouseover="mover('3')">
   PHP is usually purely interpreted, as is the case with JavaScript. However, recent PHP
   implementations perform some precompilation, at least on complex scripts, which increases
   the speed of interpretation.
 </div>
 </center> </body></html>
```

EXECUTION PROCEDURE:

1. Login into LINUX OS through root credentials
2. Start Apache web Server

   [root@localhost ~]# systemctl start httpd.service
3. Change the directory to /var/www/html

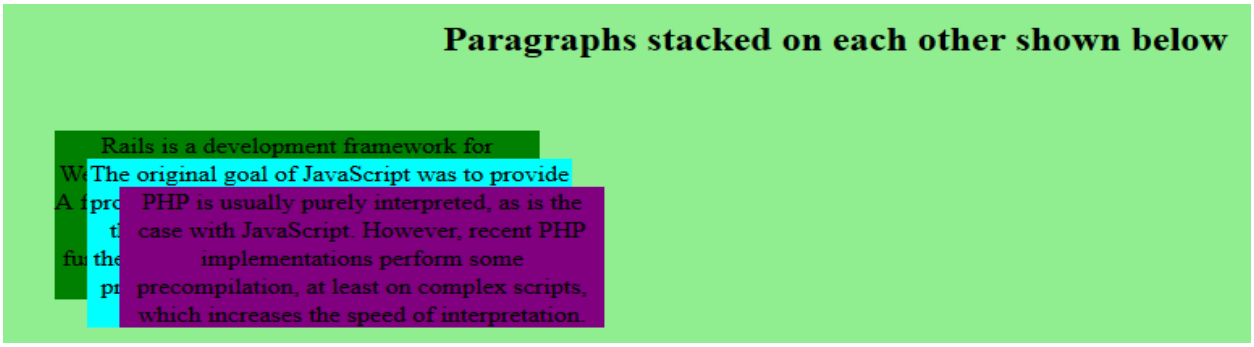
   [root@localhost ~]# cd /var/www/html
4. Type the above XHTML code and save the file.

   [root@localhost html]# gedit Labprogram3a.html
5. Go to web browser (Mozilla Firefox or Konqueror) and type in the URL as

   http://localhost/ Labprogram3a.html [Enter]

OUTPUT:

Screen 1: Three paragraphs stacked on each other.



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.

SOLUTION:

```
<?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">
<!DOCTYPE HTML PUBLIC "-//w3c//DTD XHTML 1.1//EN">
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Paragraphs stacked on each other</title>
<style type="text/css">
  .para1{width:300px;background-color:green;position:absolute;
        top:100px;left:200px;z-index:0;}
  .para2{width:300px;background-color:cyan;position:absolute;
      top:120px;left:220px;z-index:0;}
  .para3{width:300px;background-color:purple;position:absolute;
      top:140px;left:240px;z-index:0;}
 </style>
<script type="text/javascript">
var topLayer="3";
var origPos;
function mover(toTop,pos)
{
            var newTop=document.getElementById(toTop).style;
            newTop.zIndex="10";
            topLayer=document.getElementById(toTop).id;
           origPos=pos;
}

function moveBack()
{
  document.getElementById(topLayer).style.zIndex=origPos;
}
</script></head>
<body bgcolor="lightgreen">
<center>
<h1> Paragraphs stacked on each other</h1>
```

```
<div  class="para1" id="1"   onmouseover="mover('1','1')" onmouseout="moveBack()">
```
Rails is a development framework for Web-based applications that access databases. A framework is a system in which much of the more-or-less standard software parts are furnished by the framework, so they need not be written by the applications developer.
```
</div>
<div class="para2" id="2" onmouseover="mover('2','2')" onmouseout="moveBack()">
```
The original goal of JavaScript was to provide programming capability at both the server and the client ends of a Web connection. Since then, JavaScript has grown into a full-fledged programming language that can be used in a variety of application areas.
```
</div>
<div class="para3" id="3" onmouseover="mover('3','3')" onmouseout="moveBack()">
```
PHP is usually purely interpreted, as is the case with JavaScript. However, recent PHP implementations perform some precompilation, at least on complex scripts, which increases the speed of interpretation.
```
</div>
</center>
</body>
</html>
```

EXECUTION PROCEDURE:

1. Login into LINUX OS through root credentials
2. Start Apache web Server

   [root@localhost ~]# systemctl start httpd.service
3. Change the directory to /var/www/html

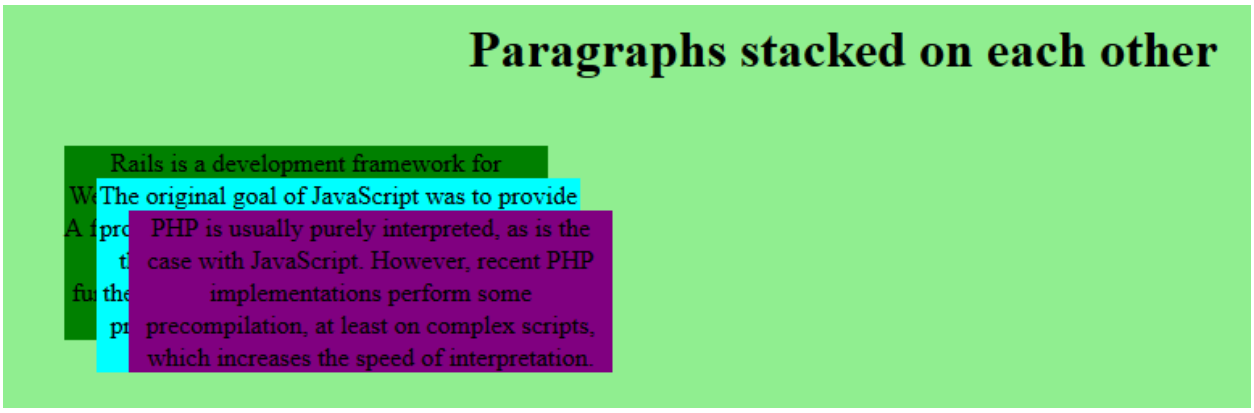
   [root@localhost ~]# cd /var/www/html
4. Type the above XHTML code and save the file.

   [root@localhost html]# gedit Labprogram3b.html
5. Go to web browser (Mozilla Firefox or Konqueror) and type in the URL as

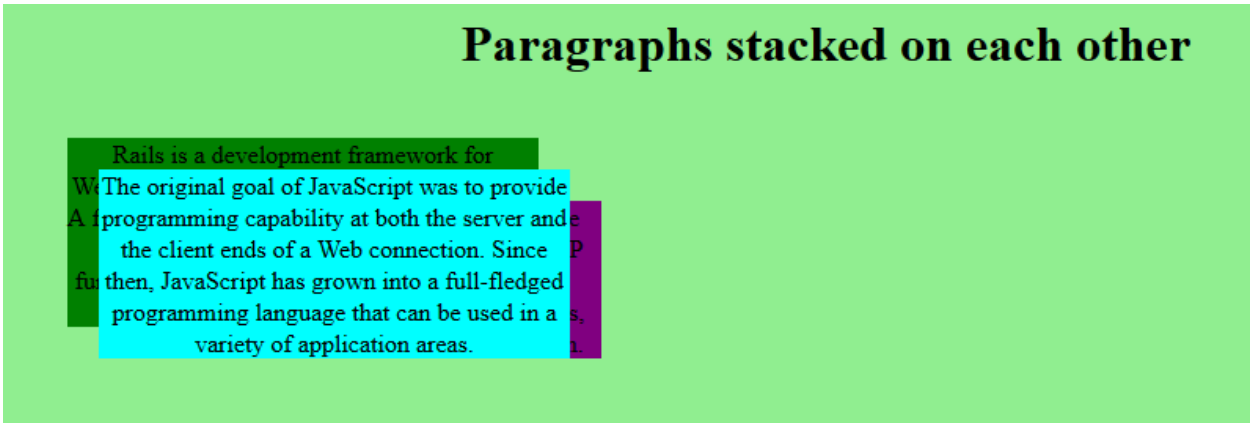
   http://localhost/ Labprogram3b.html [Enter]

OUTPUT:

Screen 1: Three paragraphs stacked on each other.



Screen 2: Paragraph is moved from the top stacking position, it returns to its original position.



4. a) Design an XML document to store information about a student in an engineering college affiliated to VTU. The information must include 100 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.

SOLUTION:

Labprogram4a.xml

```
<?xml version = "1.0" encoding ="utf-8"?>
<?xml-stylesheet type = "text/css" href = "Labprogram4a.css " ?>
<vtu>
        <student-info>
                <usn>1MV14CS001</usn>
                <name>abc</name>
                <college>SIR MVIT</college>
                <branch>CSE</branch>
                <year>2014</year>
                <email>abc@gmail.com</email>
        </student-info>
        <student-info>
                <usn>1MV14IS003</usn>
                <name>xyz</name>
                <college>SIR MVIT</college>
                <branch>ISE</branch>
                <year>2014</year>
                <email>xyz@gmail.com</email>
        </student-info>
        <student-info>
                <usn>1NC14CS001</usn>
                <name>def</name>
                <college>NCET</college>
                <branch>CSE</branch>
                <year>2014</year>
                <email>def@gmail.com</email>
        </student-info>
</vtu>
```

Labprogram4a.css

```
usn{font-family:'sans serif';color:orange;font-size:15pt;}
name{font-family:'arial';color:red;font-size:15pt;}
college{font-family:'Times New Roman';color:lime;font-size:15pt;}
branch{font-family:'Comic Sans MS';color:gray;font-size:15pt;}
year{font-family:'Century Gothic';color:blue;font-size:15pt;}
email{font-family:'Georgia';color:green;font-size:15pt;}
```

EXECUTION PROCEDURE:

1. Login into LINUX OS through root credentials
2. Start Apache web Server

   [root@localhost ~]# systemctl start httpd.service
3. Change the directory to /var/www/html

   [root@localhost ~]# cd /var/www/html
4. Type the above XML code and save the file

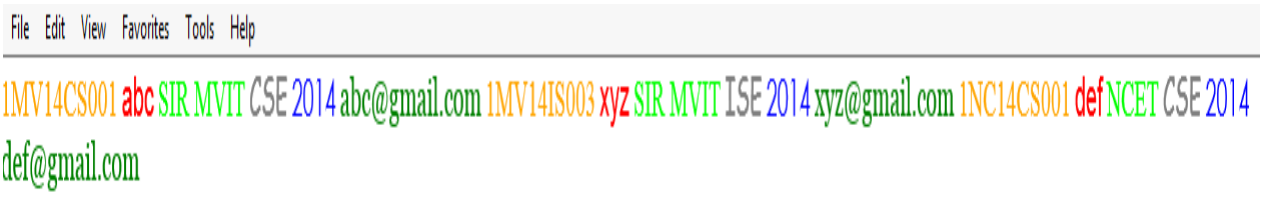
   [root@localhost html]# gedit Labprogram4a.xml
5. Type the above CSS code and save the file

   [root@localhost html]# gedit Labprogram4a.css
6. Go to web browser (Mozilla Firefox or Konqueror) and type in the URL as

   http://localhost/ Labprogram4a.xml [Enter]

OUTPUT:

Screen 1: The web page displays XML using CSS.



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

SOLUTION:

Labprogram4b.xml

```
<?xml version = "1.0" encoding ="utf-8"?>
<?xml-stylesheet type = "text/xsl" href = "Labprogram4b.xsl" ?>
<vtu>
        <student-info>
                <usn>1MV14IS001</usn>
                <name>ABC</name>
                <college>SIR MVIT</college>
                <branch>ISE</branch>
                <year>2014</year>
                <email>abc@gmail.com</email>
        </student-info>
 </vtu>
```

Labprogram4b.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 = "vtu">
<html><head><title>XML Page with XSL </title>
</head><body bgcolor="lightgreen">
<center><h2> VTU SYUDENT INFORMATION </h2>
<table border="1">
<tr><td>USN</td><td>NAME</td><td>COLLEGE</td><td>BRANCH</td><td>YEAR</td>
<td>EMAIL</td></tr>
<xsl:for-each select ="student-info">
<tr><td> <xsl:value-of select = "usn" /></td>
<td><xsl:value-of select = "name" /></td>
<td><xsl:value-of select = "college" /></td>
<td> <xsl:value-of select = "branch" /></td>
<td><xsl:value-of select = "year" /></td>
<td><xsl:value-of select = "email" /></td></tr>
</xsl:for-each></table></center>
 </body></html>
</xsl:template></xsl:stylesheet>
```

EXECUTION PROCEDURE:

1. Login into LINUX OS through root credentials
2. Start Apache web Server

   [root@localhost ~]# systemctl start httpd.service
3. Change the directory to /var/www/html

   [root@localhost ~]# cd /var/www/html
4. Type the above XML code and save the file.

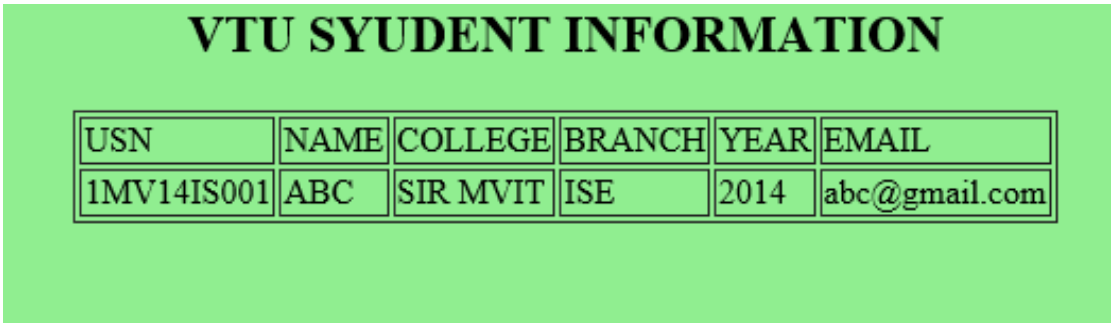
   [root@localhost html]# gedit Labprogram4b.xml
5. Type the above XSL code and save the file

   [root@localhost html]# gedit Labprogram4b.xsl
6. Go to web browser (Mozilla Firefox or Konqueror) and type in the URL as

   http://localhost/ Labprogram4b.xml [Enter]

OUTPUT:

Screen 1: The web page displays XML using XSL.



VTU SYUDENT INFORMATION

USN	NAME	COLLEGE	BRANCH	YEAR	EMAIL
1MV14IS001	ABC	SIR MVIT	ISE	2014	abc@gmail.com

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

SOLUTION:

```
#!/usr/bin/perl
print "content-type:text/html\n\n";
print "<html><head><title>Server information</title></head>";
print "<body bgcolor=lightgreen>";
print "<center><b>Server information is shown below<br><br>";
print "<table border=1><tr><td><b>ENV VARIABLE</b></td>";
print "<td><b>ENV VALUE</b></td></tr>";
foreach $a( sort keys %ENV)
{
        print "<tr><td>$a</td>";
        print "<td>$ENV{$a}</td></tr>";
}
print "</table></b></center></body></html>";
```

EXECUTION PROCEDURE:

1. Login into LINUX OS through root credentials
2. Start Apache web Server

   [root@localhost ~]# systemctl start httpd.service
3. Change the directory to /var/www/cgi-bin

   [root@localhost cgi-bin]# cd /var/www/cgi-bin
4. Type the above Perl code ad save the file.

   [root@localhost cgi-bin]# gedit Labprogram5a.pl
5. Check for errors

   [root@localhost cgi-bin]# perl Labprogram5a.pl [Enter]

6.Change the permissions

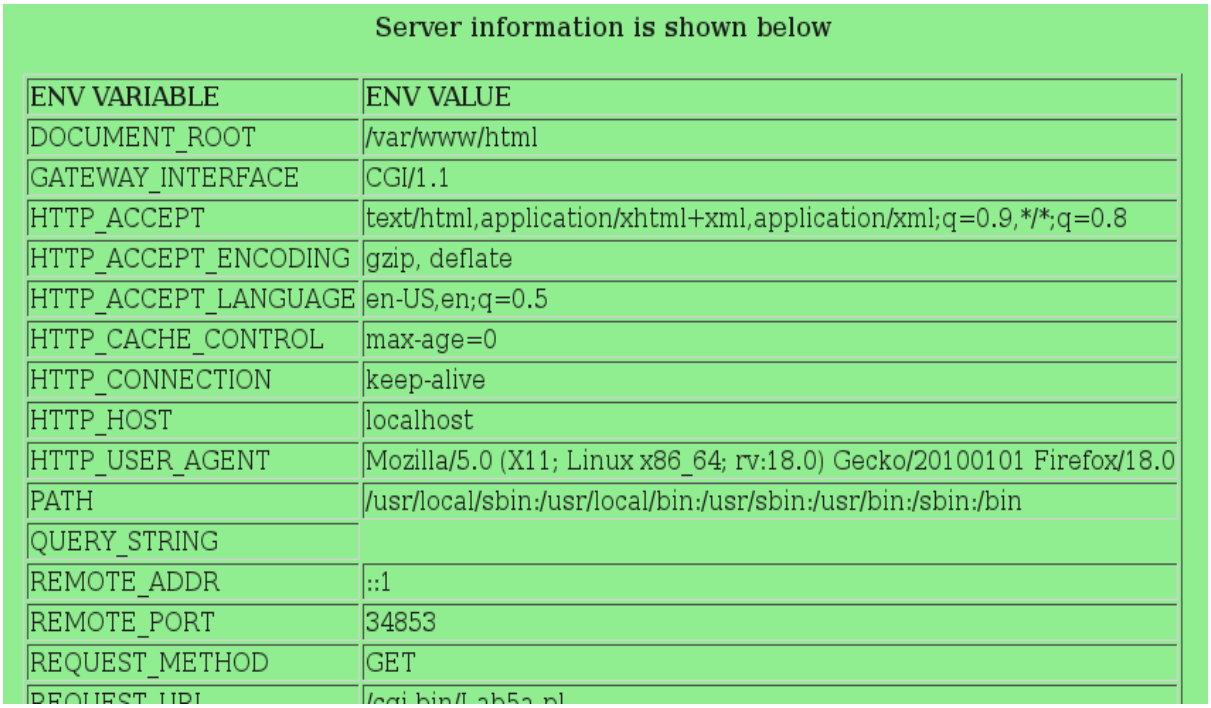
   [root@localhost cgi-bin]# chmod 777 Labprogram5a.pl [Enter]

7. Go to web browser (Mozilla Firefox or Konqueror) and type in the URL as

   http://localhost/ cgi-bin/Labprogram5a.pl [Enter]

OUTPUT:

Screen 1: The web page displays Server information.



Server information is shown below

ENV VARIABLE	ENV VALUE
DOCUMENT_ROOT	/var/www/html
GATEWAY_INTERFACE	CGI/1.1
HTTP_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_ACCEPT_LANGUAGE	en-US,en;q=0.5
HTTP_CACHE_CONTROL	max-age=0
HTTP_CONNECTION	keep-alive
HTTP_HOST	localhost
HTTP_USER_AGENT	Mozilla/5.0 (X11; Linux x86_64; rv:18.0) Gecko/20100101 Firefox/18.0
PATH	/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
QUERY_STRING	
REMOTE_ADDR	::1
REMOTE_PORT	34853
REQUEST_METHOD	GET
REQUEST_URI	/cgi-bin/Lab5a.pl

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

SOLUTION:

Lab5b.html

```
<html>
<head><title>UNIX Command Execution</title></head>
<body bgcolor="lightgreen">
<form method="get" action="http://localhost/cgi-bin/Lab5b.pl">
<center><b>ENTER THE UNIX COMMAND<br><br>
Command   <input type="text" name="cmd"></b><br><br>
<input type="submit" value="CLICK">
</centre></form>
</body></html>
```

Lab5b.pl

```
#!/usr/bin/perl
print "content-type:text/html\n\n";
use CGI;
$cgi=new CGI;
$cmd=$cgi->param("cmd");
print "<html><body bgcolor=lightgreen><center><b>";
print "output of the UNIX command executed is<br><br>";
system($cmd);
print "</b></center></body></html>";
```

EXECUTION PROCEDURE:

1. Login into LINUX OS through root credentials
2. Start Apache web Server

   [root@localhost ~]# systemctl start httpd.service
3. Change the directory to /var/www/html

   [root@localhost ~]# cd /var/www/html
4. Type the above HTML code and save the file.

   [root@localhost html]# gedit Lab5b.html
5. Change the directory to /var/www/cgi-bin

   [root@localhost cgi-bin]# cd /var/www/cgi-bin
6. Type the above Perl code and save the file.

   [root@localhost cgi-bin]# gedit Lab5b.pl
5. Check for errors

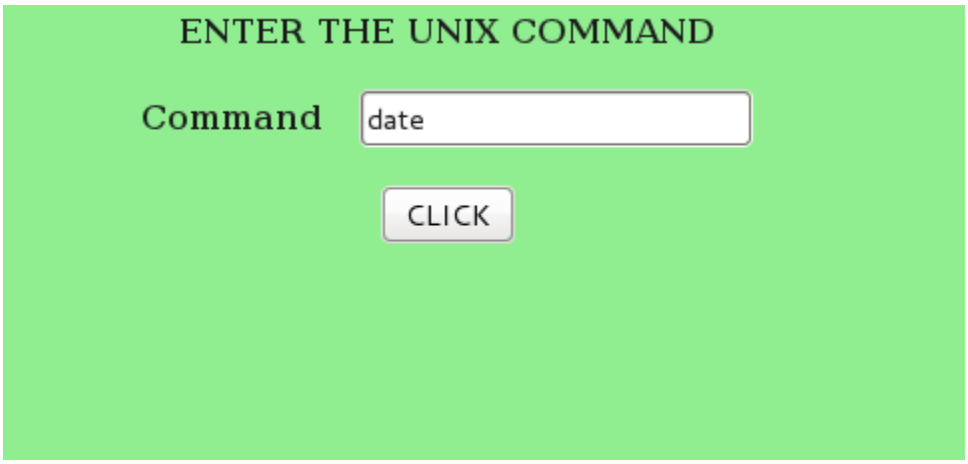
   [root@localhost cgi-bin]# perl Lab5b.pl [Enter]
6. Change the permissions

   [root@localhost cgi-bin]# chmod 777 Lab5b.pl
7. Go to web browser (Mozilla Firefox or Konqueror) and type in the URL as

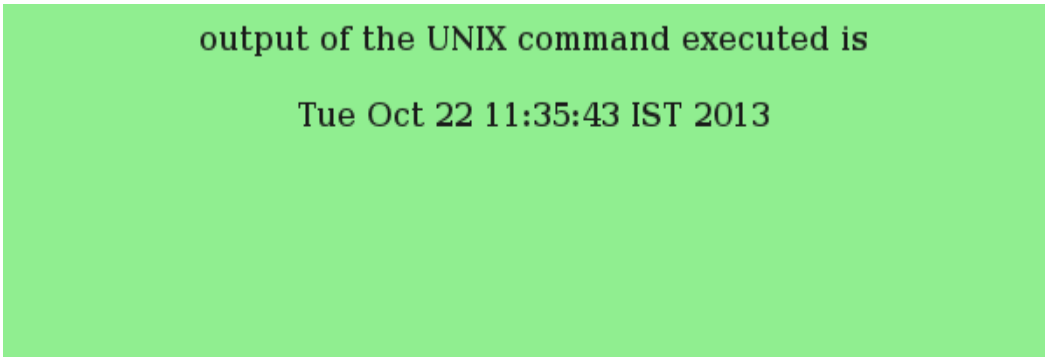
   http://localhost/Lab5b.html [Enter]

OUTPUT:

Screen 1: The web page to accept UNIX command to execute.



Screen 2: The web page displays the Output of the UNIX 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.

SOLUTION:

Lab6a.html

```
<html>
<head><title>Greeting Message</title></head>
<body bgcolor="lightgreen">
<form method="get" action="http://localhost/cgi-bin/Lab6a.pl">
<center><b>Enter your Name</b><br><br>
<b>Name</b> <input type="text" name="nam"><br><br>
<input type="submit" value="CLICK">
</center></form></body></html>
```

Lab6a.pl:

```
#!/usr/bin/perl
use CGI;
print "content-type:text/html\n\n";
$cgi=new CGI;
$usr=$cgi->param("nam");
@a=("hi","hru","welcome","wt u dng");
$greet=rand(4);
print "<html><body bgcolor=lightgreen><center><b>";
print "Hi $usr your greeting message is:<br>$a[$greet]";
print "</b></center></body></html>";
```

EXECUTION PROCEDURE:

1. Login into LINUX OS through root credentials
2. Start Apache web Server

   [root@localhost ~]# systemctl start httpd.service
3. Change the directory to /var/www/html

   [root@localhost ~]# cd /var/www/html
4. Type the above HTML code and save the file.

   [root@localhost html]# gedit Lab6a.html
5. Change the directory to /var/www/cgi-bin

   [root@localhost html]# cd /var/www/cgi-bin

6. Type the above Perl code and save the file.

   [root@localhost cgi-bin]# gedit Lab6a.pl

5. Check for errors

   [root@localhost cgi-bin]# perl Lab6a.pl [Enter]

6. Change the permissions

   [root@localhost cgi-bin]# chmod 777 Lab6a.pl

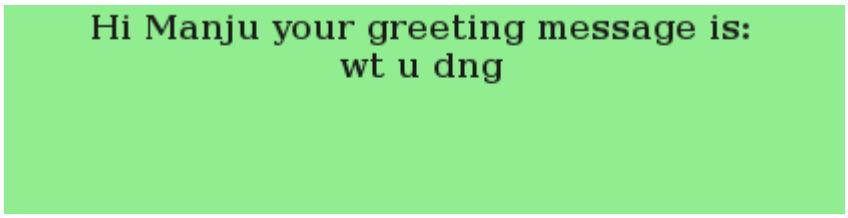
7. Go to web browser (Mozilla Firefox or Konqueror) and type in the URL as

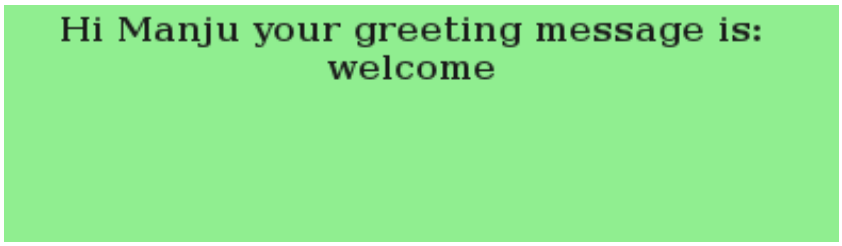
   http://localhost/Lab6a.html [Enter]

OUTPUT:

Screen 1: The web page to accept username.

Screen 2: The web page displays the greeting message





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.

SOLUTION:

Lab6b.pl

```
#!/usr/bin/perl
print "content-type:text/html\n\n";
open(IN,"<visit.txt");
$count=<IN>;
close(IN);

open(OUT,">visit.txt");
$count++;
print OUT $count;
close(OUT);

print "<html><body bgcolor=lightgreen><center><b>";
print "Number of Vistors to this page is::$count";
print "</b></center></body></html>";
```

EXECUTION PROCEDURE:

1. Login into LINUX OS through root credentials
2. Start Apache web Server

   [root@localhost ~]# systemctl start httpd.service
3. Change the directory to /var/www/cgi-bin

   [root@localhost ~]# cd /var/www/cgi-bin
4. Type the above Perl code and [save the file]

   [root@localhost cgi-bin]# gedit Lab6b.pl
5. Check for errors

   [root@localhost cgi-bin]# perl Lab6b.pl [Enter]

6.Change the permissions

   [root@localhost cgi-bin]# chmod 777 Lab6b.pl
7. Change the file permissions (visit.txt)

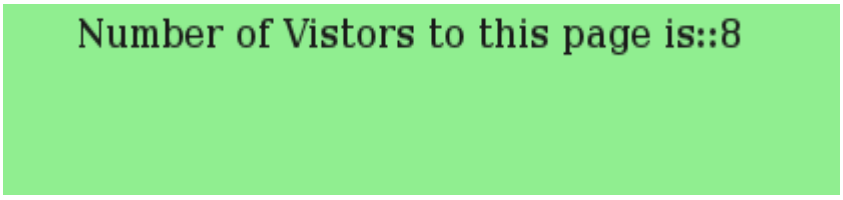
   [root@localhost cgi-bin]# chmod 777 visit.txt

8. Go to web browser (Mozilla Firefox or Konqueror) and type in the URL as

http://localhost/ cgi-bin/Lab6b.pl [Enter]

OUTPUT:

Screen 1: The web page displays the number of visitors visiting to the file. The visitors count will increase on each refresh.



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

SOLUTION:

Labprogram7.pl

```
#!/usr/bin/perl
print "content-type:text/html\n\n";
print "<html><title>Digital Clock</title><head>";
print "<META HTTP-EQUIV=refresh content=0";
print " URL=http://localhost/cgi-bin/Lab7.pl>";
($s,$m,$h)=localtime(time);
print "<body bgcolor=lightgreen><center><b>”;
print  “DIGITAL CLOCK IS SHOW BELOW<br><br>";
print "$h:$m:$s</b></center></html>";
```

EXECUTION PROCEDURE:

1. Login into LINUX OS through root credentials
2. Start Apache web Server

   [root@localhost ~]# systemctl start httpd.service
3. Change the directory to /var/www/cgi-bin

   [root@localhost ~]# cd /var/www/cgi-bin

4. Type the above Perl code and save the file.

   [root@localhost cgi-bin]# gedit Labprogram7.pl

5. Check for errors

   [root@localhost cgi-bin]# perl Labprogram7.pl [Enter]

6.Change the permissions

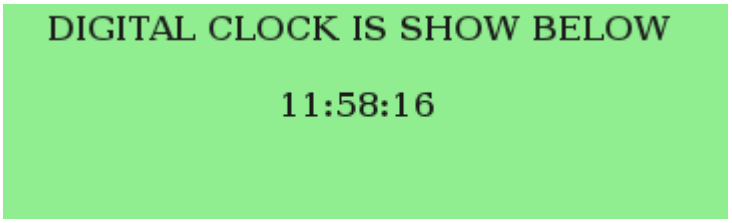
   [root@localhost cgi-bin]# chmod 777 Labprogram7.pl

7. Go to web browser (Mozilla Firefox or Konqueror) and type in the URL as

   http://localhost/ cgi-bin/Labprogram7.pl [Enter]

OUTPUT:

Screen 1: The web page displays the digital clock.



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.

SOLUTION:

Lab8.html

```
<html>
<head><title>Name & Age Insertion</title></head>
<body bgcolor="lightgreen">
<form method="get" action="http://localhost/cgi-bin/Lab8.pl">
<center><b>Enter the following information</b><br><br>
NAME   <input type="text" name="nam"><br><br>
AGE    <input type="text" name="age"><br><br>
<input type="submit" value="insert">
</center></form></body></html>
```

Lab8.pl

```
#!/usr/bin/perl
use DBI;
use CGI;
print "content-type:text/html\n\n";
$ab=new CGI;
$nam=$ab->param("nam");
$age=$ab->param("age");
$con=DBI->connect("DBI:mysql:web","sirmvit","cse");
$res=$con->prepare("insert into age_info values('$nam','$age')");
$res->execute();

$res1=$con->prepare("select * from age_info");
$res1->execute();
$len=$res1->rows;
print "<html><body bgcolor=lightgreen><center>";
if($len <= 0)
{
        print "No rows found";
}
else
{
        print "The database table contents shown below<br><br><table border=1>";
```

```
        print "<tr><td>NAME</td><td>AGE</td></tr>";
        while(@data=$res1->fetchrow_array())
        {
                print "<tr><td>$data[0]</td><td>$data[1]</td></tr>";
        }
        print "</table>";
}
print "</center></body></html>";
```

EXECUTION PROCEDURE:

1. Login into LINUX OS through root credentials
2. Start Apache web Server

   [root@localhost ~]# systemctl start httpd.service [Enter]

3. Start Mysql Server

   [root@localhost ~]# systemctl start mysqld.service [Enter]

4. Create Database and Table to insert Name & Age information.

```
[root@localhost ~]#mysql [Enter]
mysql> create database web;
Query OK, 1 row affected (0.08 sec)
mysql> use web;
Database changed
mysql> grant all on web.* to sirmvit@localhost identified by 'cse';
Query OK, 0 rows affected (0.10 sec)
mysql> create table age_info(name varchar(20) primary key, age int);
Query OK, 0 rows affected (0.08 sec)
mysql> commit;
Query OK, 0 rows affected (0.00 sec)
mysql>exit
```

5. Change the directory to /var/www/html

   [root@localhost ~]# cd /var/www/html

6. Type the above HTML code and save the file.

   [root@localhost html]# gedit Lab8.html

7. Change the directory to /var/www/cgi-bin

   [root@localhost cgi-bin]# cd /var/www/cgi-bin

6. Type the above Perl code and save the file.

   [root@localhost cgi-bin]# gedit Lab8.pl

5. Check for errors

   [root@localhost cgi-bin]# perl Lab8.pl [Enter]

6. Change the permissions

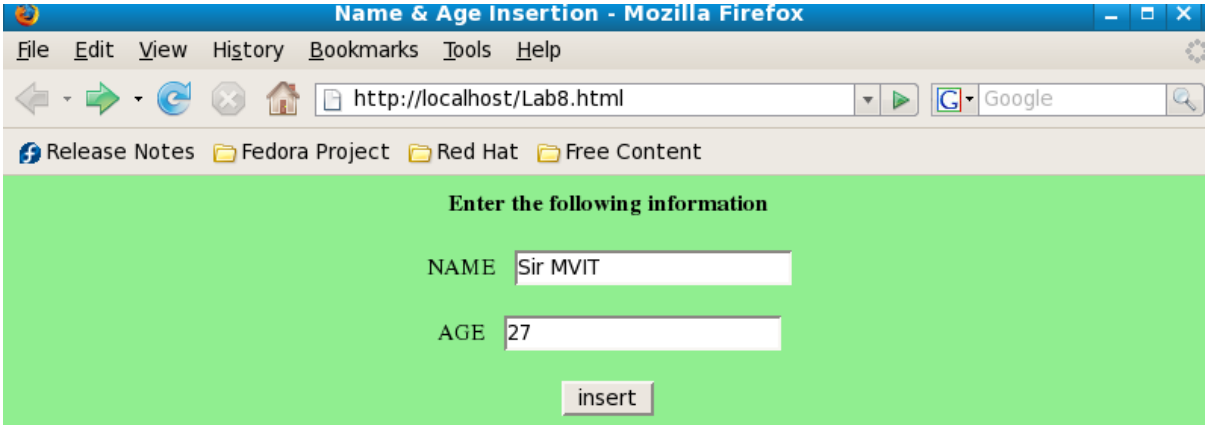
   [root@localhost cgi-bin]# chmod 777 Lab8.pl

7. Go to web browser (Mozilla Firefox or Konqueror) and type in the URL as

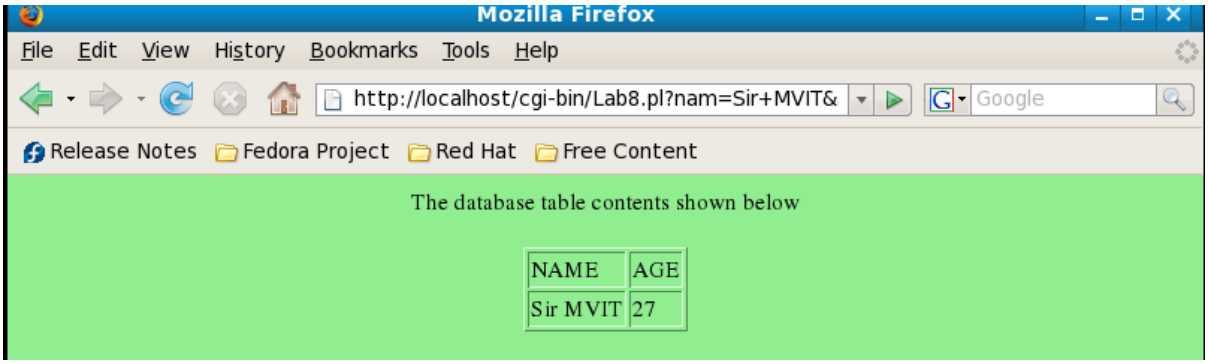
   http://localhost/Lab8.html [Enter]

OUTPUT:

Screen 1: The web page to accept Name and Age information.



Screen 2: The web page to display Database table contents.



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.

SOLUTION:

```
<?php

      $expire_days=60*60*24*30+time();
      setcookie("visitdate",date("G:i:s a - m/d/y"),$expire_days);
      if(isset($_COOKIE["visitdate"]))
      {
            $lastvisit=$_COOKIE["visitdate"];
            print "<center><br><b> Last visited on:$lastvisit";
      }
      else
      {
            print "please set the cookie";
      }
         print "</b></center>";
?>
```

EXECUTION PROCEDURE:

1. Login into LINUX OS through root credentials
2. Start Apache web Server

   [root@localhost ~]# systemctl start httpd.service
3. Open new terminal & change the directory to /var/www/html

   [root@localhost ~]# cd /var/www/html
4. Type the above PHP code and save the file.

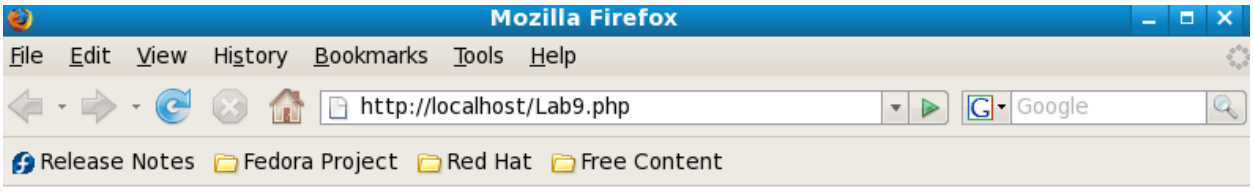
   [root@localhost html]# gedit Lab9.php
5. Check for errors

   [root@localhost html]# php Lab9.php [Enter]
6. Go to web browser (Mozilla Firefox or Konqueror) and type in the URL as

   http://localhost/Lab9.php [Enter]

OUTPUT:

Screen 1: The web page displays the 'Last visited on' date-time.



Last visited on:2:56:25 am - 10/22/13

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.

SOLUTION:

```
<?php
        session_start();
        session_register("pageviewcount");
        if(!isset($_SESSION["pageviewcount"]))
        {
                $_SESSION["pageviewcount"]=0;
        }
        else
        {
                $_SESSION["pageviewcount"]++;
        }
        print "<center><b>The number of page views count:";
        print "_SESSION[pageviewcount]</b></center>";
?>
```

EXECUTION PROCEDURE:

1. Login into LINUX OS through root credentials
2. Start Apache web Server

   [root@localhost ~]# systemctl start httpd.service

3. Change the directory to /var/www/html

   [root@localhost html]# cd /var/www/html

4. Type the above PHP code and save the file.

   [root@localhost html]# gedit Lab10.php

5. Check for errors

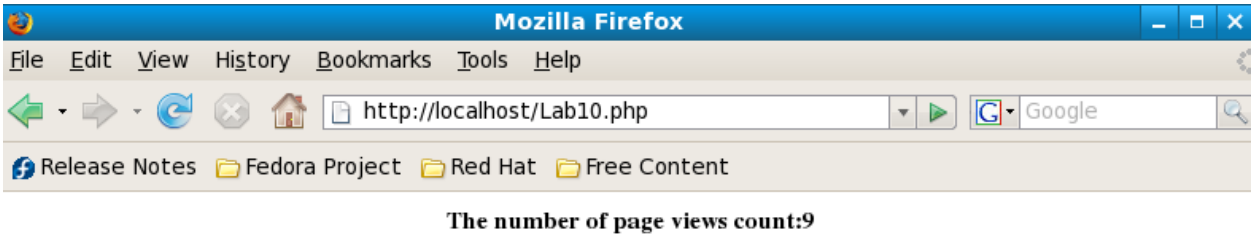
   [root@localhost html]# php Lab10.php [Enter]

6. Go to web browser (Mozilla Firefox or Konqueror) and type in the URL as

   http://localhost/Lab10.php [Enter]

OUTPUT:

Screen 1: The web page displays the page views count on web page. The count will increase on each refresh.



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.

SOLUTION:

Lab11.html

```
<?xml version="1" 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">
<head><title>Main Form</title></head>
<body bgcolor="lightgreen">
<center><b>SELECT THE FOLLOWING OPTION</b><br /><br />
<a href="Lab11a.html">CLICK HERE TO INSERT DATA INTO DATABASE
TABLE</a><br /><br />
<a href ="Lab11b.html">CLICK HERE TO SEARCH FOR A NAME</a>
</center></body></html>
```

Lab11a.html

```
<?xml version="1" 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">
<head><title>Insertion Page</title></head>
<body bgcolor="lightgreen">
<form method="get" action="insert.php">
<center><b>Enter the following Information</b><br /><br />
<table border="0">
        <tr><td>NAME </td><td><input type="text" name="nam"></td></tr>
        <tr><td>ADDR LINE1 </td><td><input type="text" name="add1"></td></tr>
        <tr><td>ADDR LINE2 </td><td><input type="text" name="add2"></td></tr>
        <tr><td>EMAIL  </td><td><input type="text" name="email"></td></tr>
</table><br />
<input type="submit" value="INSERT">
</center></form>
</body></html>
```

Lab11b.html

```
<?xml version="1" 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">
<head><title>Search for a Name</title></head>
<body bgcolor="lightgreen">
<form method="get" action="search.php">
<center><b>Enter the Name <br /><br />
Name   <input type="text" name="nam"></b><br /><br />
<input type="submit" value="SEARCH">
</center></form></body></html>
```

insert.php

```
<?php
        $name=$_GET["nam"];
        $add1=$_GET["add1"];
        $add2=$_GET["add2"];
        $email=$_GET["email"];
        mysql_connect("localhost","sirmvit","ise");
        mysql_select_db("address");
        mysql_query("insert into person_info values('$name','$add1','$add2','$email')");
        $res=mysql_query("select * from person_info");
        $len=mysql_num_rows($res);
        print "<html><body bgcolor=lightgreen><center>";
        if($len>0)
        {
                print "<b>The Database table contents shown below<br><br>";
                print "<table border=1><tr><td>Name</td><td>Address 1</td>";
                print "<td>Address 2</td><td>Email</td></tr>";
                while($data=mysql_fetch_row($res))
                {
                        print "<tr><td>$data[0]</td>";
                        print "<td>$data[1]</td>";
                        print "<td>$data[2]</td>";
                        print "<td>$data[3]</td></tr>";
                }
                print "</table>";
        }
```

```
        else
        {
                print "no rows found";
        }

        print "</b></center></body></html>";
        mysql_close($con);
?>
```

search.php

```
<?php

        $nam=$_GET["nam"];
        $con=mysql_connect("localhost","sirmvit","ise");
        mysql_select_db("address");
        $res=mysql_query("select * from person_info where name='$nam'");
        print "<html><body bgcolor=lightgreen><center><b>";
        if(mysql_num_rows($res) >0)
        {
                print "The Searched Results shown below<br><br>";
                print "<table border=1><tr><td>NAME</td><td>ADDRESS 1</td>";
                print "<td>ADDRESS 2</td><td>EMAIL</td></tr>";

                while($a=mysql_fetch_row($res))
                {
                print"<tr><td>$a[0]</td>";
                print"<td>$a[1]</td>";
                print "<td>$a[2]</td>";
                print "<td>$a[3]</td></tr>";
                }
                print "</table>";
        }
        else
        {
                print "No rows Found";
        }
        print "</b></center></body></html>";

?>
```

EXECUTION PROCEDURE:

1. Login into LINUX OS through root credentials
2. Start Apache web Server

   [root@localhost ~]# systemctl start httpd.service [Enter]

3. Start Mysql Server

   [root@localhost ~]# systemctl start mysqld.service [Enter]

4. Create Database and Table to insert address information.

```
[root@localhost ~]#mysql [Enter]
mysql> create database address;
Query OK, 1 row affected (0.08 sec)
mysql> use address;
Database changed
mysql> grant all on address.* to sirmvit@localhost identified by 'ise';
Query OK, 0 rows affected (0.10 sec)
mysql> create table person_info(name varchar(20),add1 varchar(30),add2
        varchar(30),email varchar(20) primary key);
Query OK, 0 rows affected (0.08 sec)
mysql> commit;
Query OK, 0 rows affected (0.00 sec)
mysql>exit
```

5. Change the directory to /var/www/html

   [root@localhost ~]# cd /var/www/html

6. Type the above HTML code

   [root@localhost html]# gedit Lab11.html [save the file]

   [root@localhost html]# gedit Lab11a.html [save the file]

   [root@localhost html]# gedit Lab11b.html [save the file]

5. Type the above PHP code

   [root@localhost html]# gedit insert.php [save the file]

   [root@localhost html]# php insert.php [check for errors]

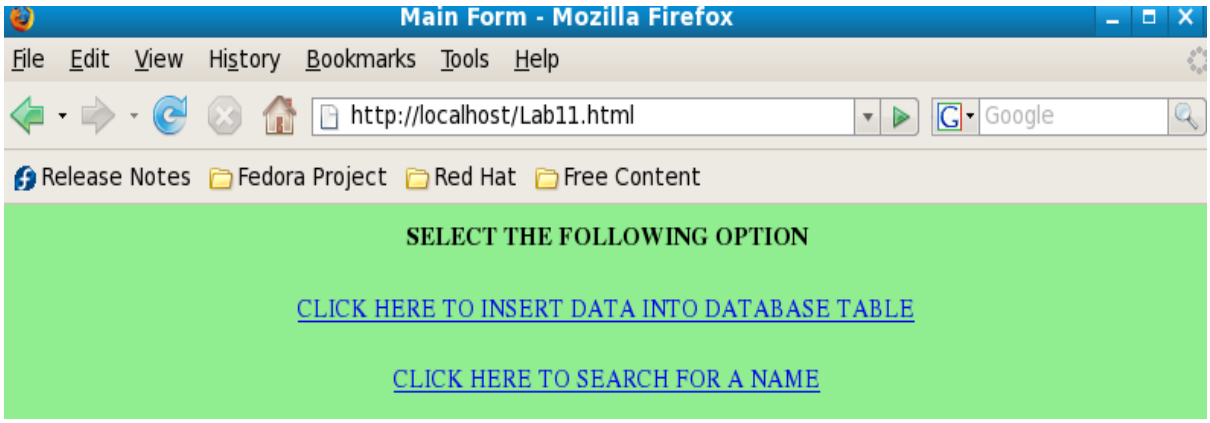
   [root@localhost html]# gedit search.php [save the file]

   [root@localhost html]# php search.php [check for errors]

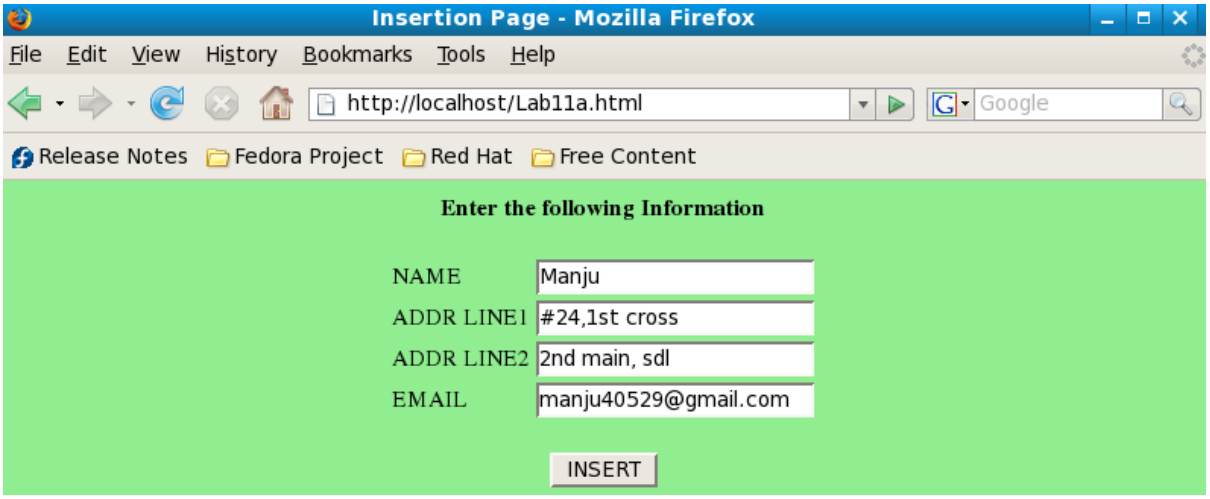
6. Go to web browser (Mozilla Firefox or Konqueror) and type in the URL as
   http://localhost/Lab11.php [Enter]

OUTPUT:

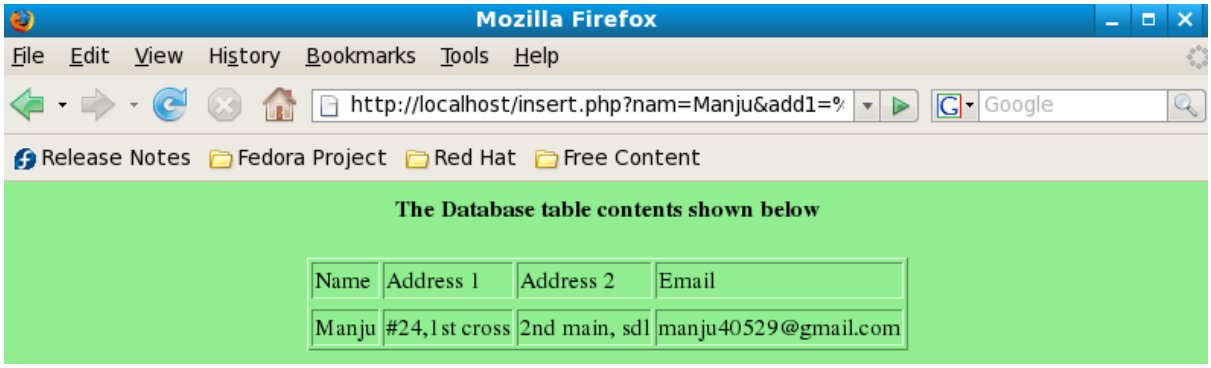
Screen 1: The web page to select insert or search option.



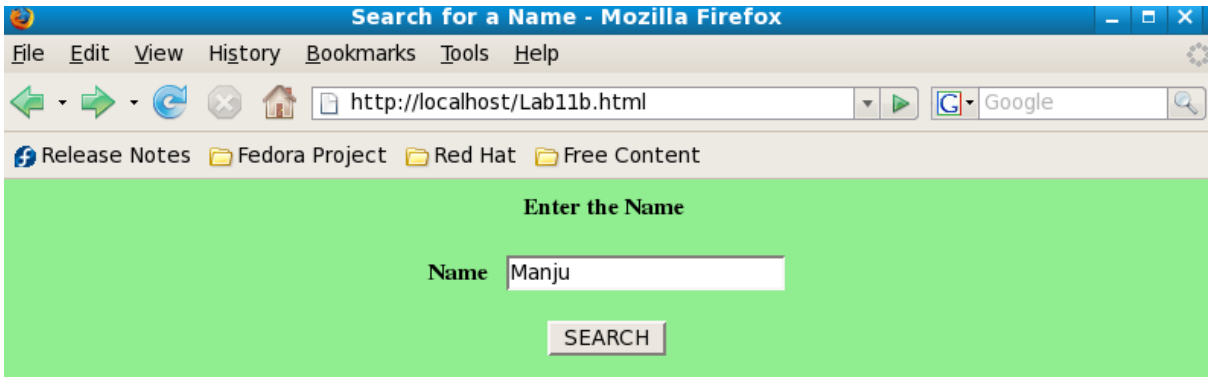
Screen 2: The web page to enter person information.



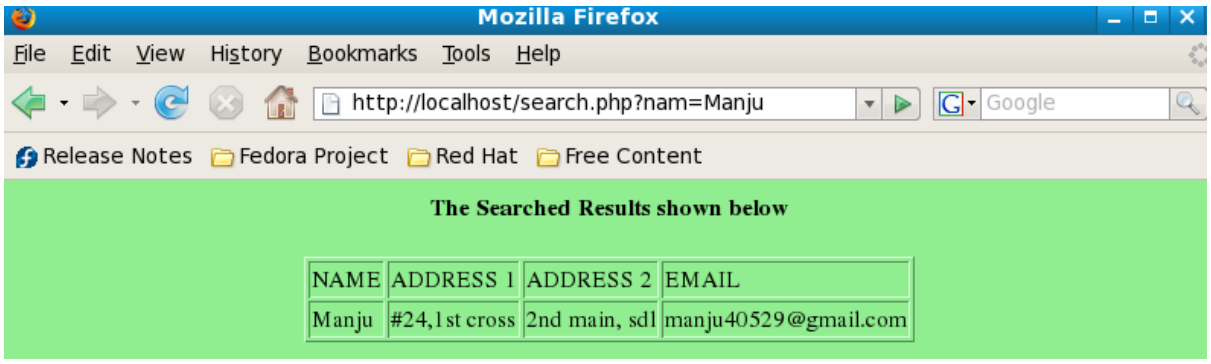
Screen 3: The web page to show inserted data.



Screen 4: The web page to enter name to search be searched.



Screen 5: The web page to displays the searched results.



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.

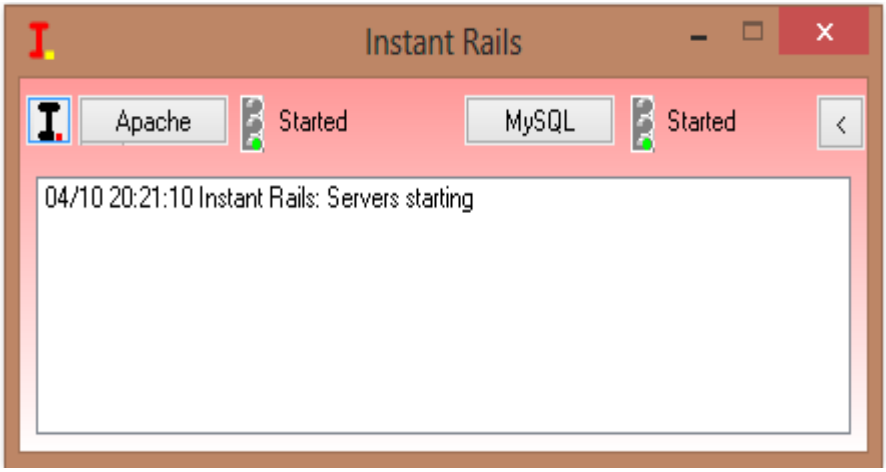
SOLUTION:

Installation:

Step 1: download InstantRails zip file from **www.rubyforge.org/projects/instantrails** for windows.

Step 2: Unzip InstantRails where you want it to reside.

Step 3: Start InstantRails: This is done by clicking the thick **I**, which is the icon of the Instant Rails application, in the directory in which Instant Rails was installed. A small window then opens, as shown in the following figure.



<u>Steps to create rails Application:</u>

Step 1: Open a Ruby console window through Instant Rails by clicking the black "I" button Rails Applications -> Open Ruby Console Window. This entry opens a command-line window in the **rails_apps** subdirectory of the directory in which Instant Rails was installed.

Step 2: Create new rails application in the **rails_apps** directory with the following command.

>rails -d mysql labprogram12

Where labprogram12, name of the rails application. Next is move to the labprogram12 directory.

Step 3: Configure the database: It is customary in Rails applications to use three copies of the database: one for development, one for testing, and one for production. The configurations of three databases are constructed with the following single command:

labprogram12>rake db:create:all

The Rails response to this command is

(in C:/InstantRails-2.0-win/rails_apps/labprogram12)

Step 4: Create Database Table: The following command is used to create database table with five columns: acct, of integer type; title, of string type; author, of string type; edition, of string type; publication, of string type.

labprogram12>ruby script/generate scaffold bookinfo acct:integer title:string author:string edition:string pub:string

Step 5: Create migration: the following command is used to migrate database table.

labprogram12>rake db:migrate

The Rails response to this command is

(in C:/InstantRails-2.0-win/rails_apps/labprogram12)

== 1 CreateBookinfos: migrating ==========

-- create_table(:bookinfos)

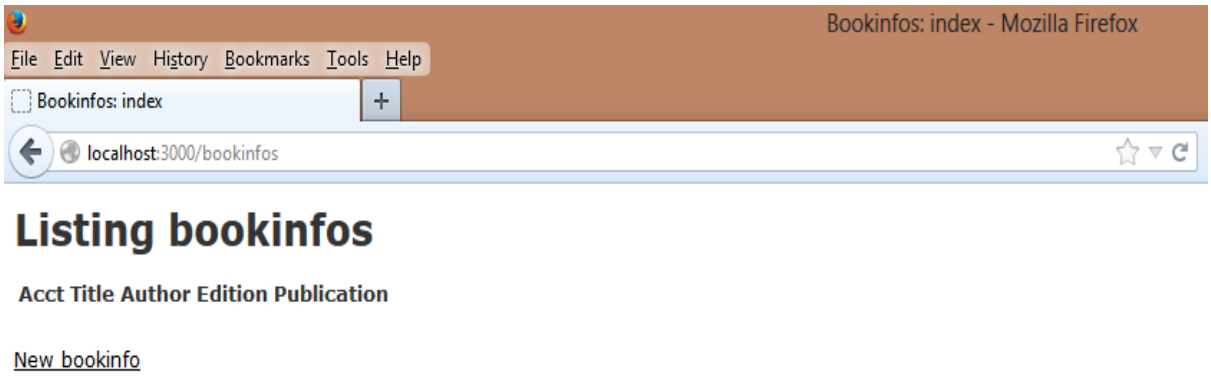
-> 0.0630s

== 1 CreateBookinfos: migrated (0.0630s) ==========

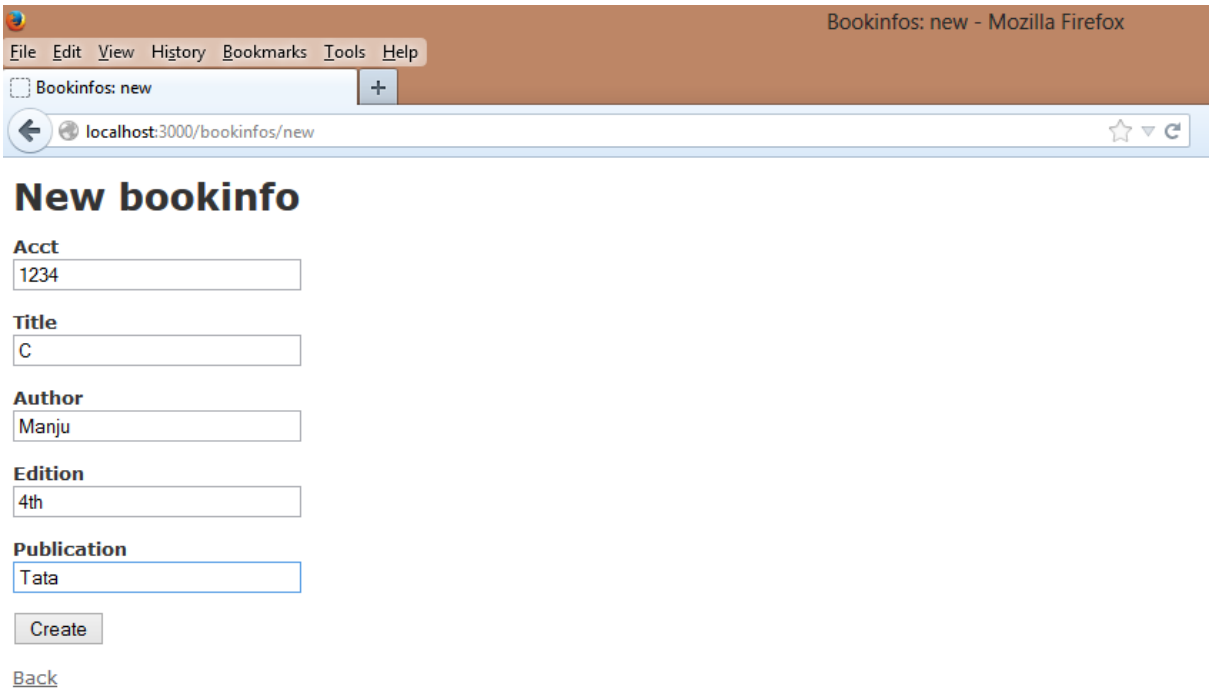
Step 6: Start WEBrick Server: the following command is used to start WEBrick server to test Rails application.

labprogram12>ruby script/server

Step 7: Go to browser and type in the URL as **http://localhost:3000/bookinfos**, we will get a Display shown below.



Step 8: Insert book information by clicking on New bookinfo. Click Create button after entering the fields.



Step 9: Press CTRL+C in command prompt (Ruby console window) and type the following command to create new controller called main.

labprogram12>ruby script/generate controller main [Enter]

the above command will create a file called main_controller.rb in C:\InstantRails-2.0-win\rails_apps\labprogram12\app\controllers

Step 10: Create a file called search.html.erb file in C:\InstantRails-2.0-win\rails_apps\ labprogram12\app\views\main with the following content.

```
<html>
<head><title>Search Book Information</title></head>
<body bgcolor="lightgreen">
<form action="search_book">
<center><b>Enter the Book Title<br><br>
Title <input type="text" name="title"><br><br>
<input type="submit" value="submit">
</b></center></form></body></html>
```

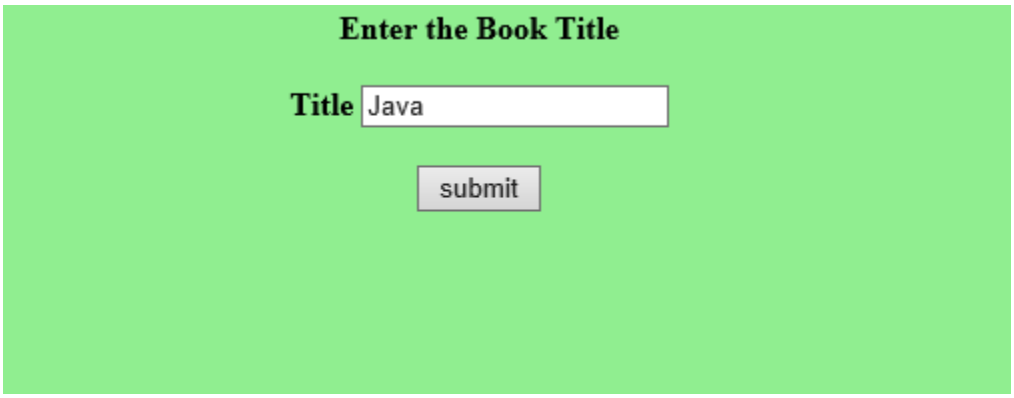
Step 11: Open the file called main_controller.rb (C:\InstantRails-2.0-win\rails_apps\ labprogram12\app\controllers) and type the following code and save the file.

```
class MainController < ApplicationController
def search_book
@title=params[:title]
@book_search= Bookinfo.find(:all,:conditions=>["title=?",@title])
end
end
```

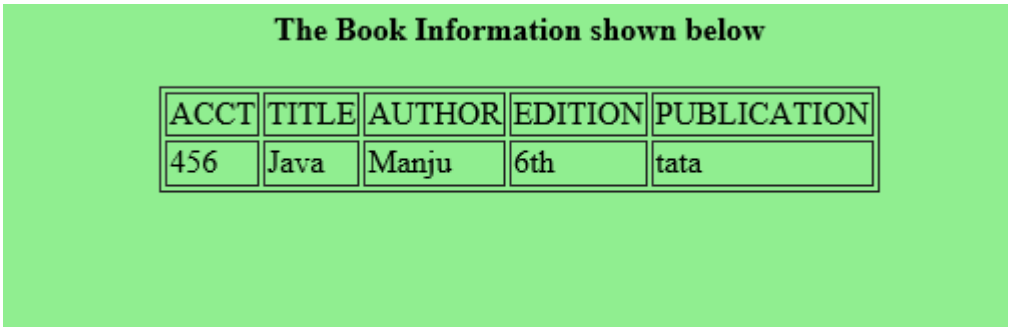
Step 12: Create a file called search_book.html.erb file in C:\InstantRails-2.0-win\rails_apps\ labprogram12\app\views\main with the following content.

```
<html>
<head><title>Searched Book Information</title></head>
<body bgcolor="lightgreen">
<center><b>
<% if @book_search.length == 0 %>
        No Rows found
<% else %>
The Book Information shown below<br><br>
<table border=1>
<tr><td>ACCT</td><td>TITLE</td> <td>AUTHOR</td>
<td>EDITION</td><td>PUBLICATION</td></tr>
<% @book_search.each do |book| %>
<tr>
      <td><%=book.acct%></td>
      <td><%=book.title%></td>
      <td><%=book.author%></td>
      <td><%=book.edition%></td>
      <td><%=book.pub%></td>
</tr>
<% end %>
</table>
<% end %>
</b></center>
</body>
</html>
```

Step 13: Go to browser and type in the URL as **http://localhost:3000/main/search**, we will get a Display shown below.



Step 14: The web page displays the searched results shown as below.



ADDITIONAL PROGRAMS

1. Develop and demonstrate a XHTML document that illustrates the use external style sheet, ordered list, table, borders, padding, color, and the tag.

SOLUTION:

external.css

```
table{font-family:times new roman;
        font-size:10pt;
        border-style:solid;
        border-color:red;}
 td{ padding:10pt;}
 li{list-style-type:lower-roman;
        font-family:times new roman;
        font-size:14pt;
        font-weight:bold;color:gray;}
body{background-color:lightgreen;}
p{color:blue;}
span{text-decoration:underline;color:orange;}
```

demo.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">
<head>
<link rel="stylesheet" type="text/css" href=" external.css" />
<title> Demonstrate external style sheets </title>
</head><body>
<h3>order list demo shown below:</h3>
<ol>
        <li>web lab</li><li>cn lab</li>
</ol>
<h3>table, border and padding demo shown below :</h3>
<table border="1"><tr><td>USN</td><td>NAME</td></tr>
<tr><td>VITcs123</td><td>XYZ</td></tr></table><br />
<h3>color demo shown below:</h3>
<p>web programming lab</p>
<h3>span demo shown below :</h3>
Welcome to <span> web</span> programming lab</body></html>
```

2. Develop and demonstrate a XHTML file that includes JavaScript script that uses functions for the following problem:
Parameter: A string Output: The position in the string of the left-most vowel

SOLUTION:
vowel.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">
<head><title>Search for vowel</title>
<script type="text/javascript">
function findpos()
{
     var str=prompt("Enter the String:","");
     var pos=str.search(/a|e|i|o|u|A|E|I|O|U/);
     var len=str.length;
     alert("length of the string is:"+len);
     if(pos >= 0)
          alert("Position of the leftmost vowel is" +pos);
   else
     alert("vowel is not found ");
}
</script>
</head>
<body bgcolor="lightgreen">
<center> Click here to enter the  String :  <input type="submit" value="CLICK!"
onClick="findpos()" />
</center></body></html>
```

3. Develop and demonstrate a XHTML file that includes JavaScript script that uses functions for the following problem:
Parameter: A number
Output: The number with its digits in the reverse order.

SOLUTION:
reverse.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">
<head>
<script type="text/javascript">
function reverse(number)
{
        var alphaExp = /^[0-9]+$/;
        if(!number.value.match(alphaExp))
        {
                alert("Input should be positive numeric");
                return false;
        }
         var rn=0, n= number.value;
         while(n!=0)
        {
                  r = n%10;
                  n = Math.floor(n/10);
                  rn = rn*10 + r;
        }
         alert("The " + number.value + " in reverse is " + rn);
}
</script>
</head>
<body bgcolor="lightgreen"><br />
<center>Enter a number :
<input type=text  name="no"><br /><br />
<input type="button" value="REVERSE" onclick="reverse(no)"/ >
</center></body>
</html>
```

4. Using PHP and MySQL, develop a program 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.

SOLUTION:

main.html :

```
<html>
<head><title>Main form</title></head>
<body bgcolor="lightgreen">
<center><b>SELECT THE FOLLOWING OPTION</b><br /><br />
<a href="insert.html">CLICK HERE TO INSERT INTO DATABASE TABLE</a><br /><br />
<a href ="search.html">CLICK HERE TO SEARCH FOR A BOOK </a>
</center></body></html>
```

insert.html :

```
<html>
<head><title>insert into database</title></head>
<body bgcolor="lightgreen">
<form method="get" action="insert.php">
<center><b>ENTER THE FOLLWING FIELDS</b><br><br>
<table>
<tr><td>ACC NO</td> <td><input type="text" name="nam"></td></tr>
<tr><td>TITLE</td> <td> <input type="text" name="title"></td></tr>
<tr><td>AUTHOR</td><td> <input type="text" name="author"></td></tr>
<tr><td>EDITION</td><td><input type="text" name="edition"></td></tr>
<tr><td>PUBLICATION</td><td> <input type="text" name="pub"></td></tr></table>
<br><br>
<input type="submit" value="INSERT">
</center></form></body></html>
```

search.html :

```
<html>
<head><title>Search Page</title></head>
<body bgcolor="lightgreen">
<form method="get" action="search.php">
<center><b>ENTER THE TITLE OF THE BOOK TO BE SEARCH</b><br><br>
TITLE OF THE BOOK  <input type="text" name="nam"><br><br>
<input type="submit" value="SEARCH">
</center></form></body></html>
```

insert.php :

```
<html><body bgcolor="lightgreen"><center>
<b>DATABASE CONTENTS ARE SHOWN BELOW</b><br><br>
<?php
$acc=$_GET["nam"];
$title=$_GET["title"];
$author=$_GET["author"];
$ed=$_GET["edition"];
$pub=$_GET["pub"];
$con=mysql_connect("localhost","dbuser","dbpwd");
mysql_select_db("web");
$res=mysql_query("insert into book_info values('$acc','$title','$author','$ed','$pub')");
?>
<table border=1>
<tr><td>ACC NO</td><td>TITLE </td> <td>AUTHOR
</td><td>EDITION</td><td>PUBLICATION</td></tr>
<?php
$res1=mysql_query("select * from book_info");
while($a=mysql_fetch_row($res1))
{
      print"<tr><td>$a[0]</td>";
      print"<td>$a[1]</td>";
      print "<td>$a[2]</td>";
      print "<td>$a[3]</td>";
         print "<td>$a[4]</td></tr>";
}
?>
</table><br><a href="main.html">CLICK HERE TO GO HOME</a></center></body></html>
```

search.php :

```
<html><body bgcolor="lightgreen"><center>
<b>DATABASE CONTENTS ARE SHOWN BELOW</b><br><br>
<?php
$title=$_GET["nam"];
$con=mysql_connect("localhost","dbuser","dbpwd");
mysql_select_db("web");
$res=mysql_query("select * from book_info where title='$title'");
?>
<?php
if(mysql_num_rows($res) >0)
{
  print “<table border=1>”;
  print “<tr><td>ACCT NO</td><td>TITLE</td>”;
  print “<td>AUTHOR</td><td>EDITION</td>”;
  print “<td>PUBLICATION</td></tr>”;

while($a=mysql_fetch_row($res))
{
        print"<tr><td>$a[0]</td>";
        print"<td>$a[1]</td>";
        print "<td>$a[2]</td>";
        print "<td>$a[3]</td>";
        print "<td>$a[4]</td></tr>";
}
}
else
{
        print "no rows found";
}
?>
</table>
<a href="main.html">CLICK HERE TO GO HOME</a></center>
</body>
</html>
```