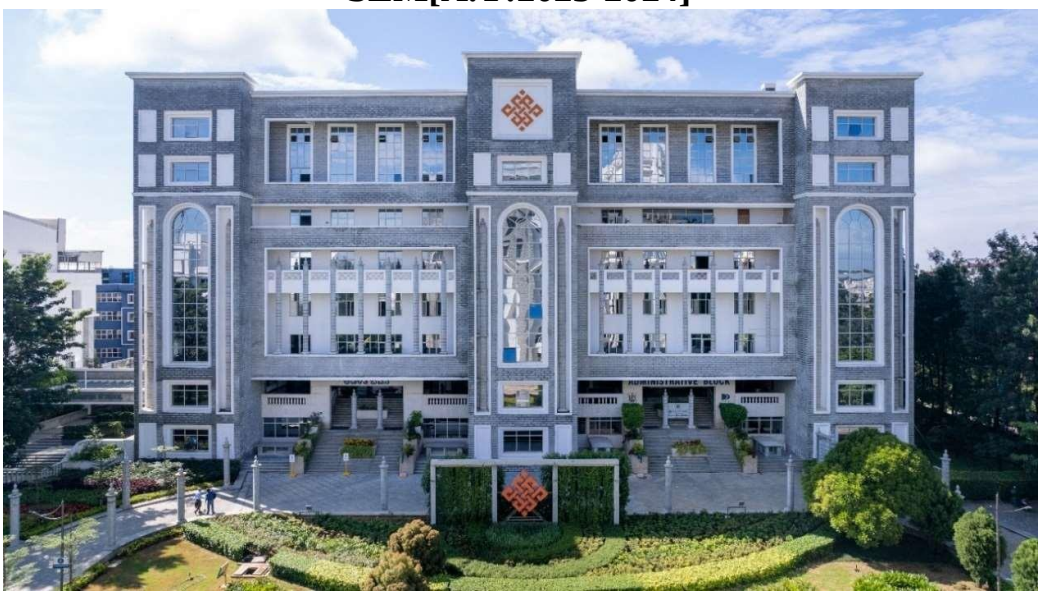
REVA UNIVERSITY
Bengaluru, India

SCHOOL OF COMPUTING AND INFORMATIONTECHNOLOGY

WEB TECHNOLOGY APPLICATIONS LAB MANUAL
[B20EJ0504]
B.TECH. III YEAR – V
SEM[A.Y:2023-2024]



Fifth Semester

B. Tech in CIT [CSSE]

INDEX

		takes student details such as Name, branch, Semester, University, date of admission, mobile number, email id and check for validity or correctness of the input data by writing a JavaScript to validate these fields.	
	Experiment 3	Clients interact with servers by sending service requests that contain input required to complete the requested task or service. Input required for requested service may be collected through a web page, that acts as an interface between users and the server, in the form of text fields, text areas, radio buttons, push buttons and so on. Hence it is better to instruct or help clients to input correct data through web page by displaying appropriate error messages or alerts as and when users supply wrong input using event handlers. To demonstrate this task, design and develop a web page using JavaScript, XHTML that collects the SRN (Valid format is: Any letter followed by two digits, followed by two letters then followed by three digits). Include event handler for the form elements that collects information to validate the input. Messages must be produced in the alert windows as and when errors are detected.	**44**
	Experiment 4	Dynamic web content is the information that is retrieved from one or more web servers depending upon what information client have requested for, and composed in response to users' requests. Advanced web technologies play a vital role in storage, processing and retrieval of dynamic web content from web servers. Hence it is important to use advanced web technologies such as XML to improve the efficiency in data retrieval. Create and save XML document for students information and display the same using cascaded style sheet.	**46**
	Experiment 5	Information technology has become part and parcel of humanity to such an extent that people can shop anything online, from anywhere, at any time using an electronic device that has access to Internet. This has brought in the concept of virtual stores which provide products at less cost. To improve sales it is mandatory to organize items catalog based on item name, item price, and manufacturer so on. For such online shopping sites, look and feel is an obvious requirement which can be achieved using CSS & XSLT. Design a document using CSS and XSLT to create a catalog of items for an online electronic shopping.	**48**

VISION OF THE UNIVERSITY

- "REVA University aspires to become an innovative university by developing excellent human resources with leadership qualities, ethical and moral values, research culture and innovative skills through higher education of global standards".

MISSION OF THE UNIVERSITY

- To create excellent infrastructure facilities and state-of-the-art laboratories and incubation centers
- To provide student-centric learning environment through innovativepedagogy and education reforms
- To encourage research and entrepreneurship through collaborations andextension activities
- To promote industry-institute partnerships and share knowledge forinnovation and development
- To organize society development programs for knowledge enhancementin thrust areas
- To enhance leadership qualities among the youth and enrich personality traits, promote patriotism and moral values

Program: B.Tech. in CSIT [Computer Science and System Engineering]

VISION OF THE SCHOOL

- REVA University aspires to become an innovative university by developing excellent human resources with leadership qualities, ethical and moral values, research culture and innovative skills through higher education of global standard.

MSSION OF THE SCHOOL

- To create excellent infrastructure facilities and state-of-the-art laboratories and incubation centers
- To provide student-centric learning environment through innovativepedagogy and education reforms
- To encourage research and entrepreneurship through collaborations andextension activities
- To promote industry-institute partnerships and share knowledge forinnovation and development
- To organize society development programs for knowledge enhancementin thrust areas
- To enhance leadership qualities among the youth and enrich personality traits, promote patriotism and moral values.

PROGRAMME EDUCATIONAL OBJECTIVES (PEOs)

PEO1: Pursue higher education in the core or allied areas of Computer Science and Systems Engineering.

PEO2: Have technical career in the core or allied areas of Computer Science Systems Engineering or start entrepreneurial activity for the growth of the economy.

PEO3: Continue to learn and to adapt to ever changing technologies in the core or allied areas of Computer Science and Systems Engineering.

PROGRAM SPECIFIC OUTCOMES (PSOs)

On successful completion of the programme, the graduates of B.Tech. CSIT (Computer Science and System Engineering) programme will be able to:

- **PSO-1** Apply the knowledge of mathematics, Computer Science and Systems Engineering to solve complex problems in CS and SE.

. PSO-2 Analyze, design, develop solutions and conduct investigations in the domains of database, networks and security, system software and system administration.

. PSO-3 Apply appropriate techniques, use modern programming languages, tools, and packages for quality software development.

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. Design / development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multi-disciplinary environments.
12. Life- long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

1. Course Objectives:

The objectives of this course are to:

1. Understand the various steps in designing a creative and dynamic website.
2. Describe the hierarchy of objects in HTML and XML.
3. Design dynamic and interactive web pages by embedding Java Script code in HTML.
4. Illustrate the advantages and use of different types of CSS.
5. Examine the HTML. Know how to use Dynamic HTML.
6. Familiarize server-side scripting language like Perl &PHP.

2. Lab Requirements:

Following are the required hardware and software for this lab, which is available in the laboratory.

Minimum System requirements:

- Processors: Intel Atom® processor or Intel® Core™ i3 processor.
- Disk space: 1 GB.
- Operating systems: Windows* 7 or later, macOS, and Linux.
- Web Browser, Notepad.

About the Lab:

In the ever-evolving landscape of technology, web development stands as a pivotal discipline that continually shapes the way we interact with the digital world. The Web Technology Lab serves as a dynamic hub where students, enthusiasts, and professionals gather to explore, experiment, and innovate in the realm of web development. This controlled environment provides a space for hands-on learning, collaboration, and skill enhancement, all essential components for individuals looking to thrive in the digital age. The primary objective of a Web Technology Lab is to familiarize participants with the fundamental concepts and advanced techniques of web development. The lab serves as a bridge between theoretical knowledge and practical implementation, allowing individuals to grasp the intricacies of web technologies through hands-on experience. It aims to nurture creativity and problem-solving skills, enabling participants to design and develop captivating web solutions that cater to the diverse needs of today's online audience. The basics of Web application tools such as HTML, XHTML and CSS will be introduced in the lab. The course also provides knowledge about advanced research topics such as XML, Perland PHP.

3. Guidelines to Students

- Equipment in the lab for the use of student community. Students need to maintaina proper decorum in the computer lab. Students must use the equipment with care. Any damage is caused is punishable.
- Students are required to carry their observation / programs book with completed exercises while entering the lab.

- Students are supposed to occupy the machines allotted to them and are not supposed to talk or make noise in the lab. The allocation is put up on the lab noticeboard.
- Lab can be used in free time / lunch hours by the students who need to use the systems should take prior permission from the lab in-charge.
- Lab records need to be submitted on or before date of submission.
- Students are not supposed to use flash drives.

Instructions to maintain the record

- Before start of the first lab they have to buy the record and bring the record to the lab.
- Regularly (Weekly) update the record after completion of the experiment and get itcorrected with concerned lab in-charge for continuous evaluation.
- In case the record is lost inform the same day to the faculty in charge and get the newrecord within 2 days the record has to be submitted and get it corrected by the faculty.
- If record is not submitted in time or record is not written properly, the evaluation marks(5M) will be deducted.

General laboratory instructions

1. Students are advised to come to the laboratory at least 5 minutes before (to the starting time), those who come after 5 minutes will not be allowed into the lab.
2. Plan your task properly much before to the commencement, come prepared to the lab with the synopsis / program / experiment details.
3. Student should enter into the laboratory with: a. Laboratory observation notes with all the details (Problem statement, Aim, Algorithm, Procedure, Program, Expected Output, etc.,) filled in for the lab session. b. Laboratory Record updated up to the last session experiments and other utensils (if any) needed in the lab. c. Proper Dress code and Identity card.
4. Sign in the laboratory login register, write the TIME-IN, and occupy the computer system allotted to you by the faculty. 5. Execute your task in the laboratory, and record the results / output in the lab observation note book, and get certified by the concerned faculty.
6. All the students should be polite and cooperative with the laboratory staff, must maintainthe discipline and decency in the laboratory.
7. Computer labs are established with sophisticated and high end branded systems, which should be utilized properly.
8. Students / Faculty must keep their mobile phones in SWITCHED OFF mode during the lab sessions. Misuse of the equipment, misbehaviors with the staff and systems etc., will attract severe punishment.
9. Students must take the permission of the faculty in case of any urgency to go out; if anybody found loitering outside the lab / class without permission during working hours will be treated seriously and punished appropriately.
10. Students should LOG OFF/ SHUT DOWN the computer system before he/she leavesthe lab after completing the task (experiment) in all aspects. He/she must ensure the system / seat is kept properly.

Course Outcomes

Upon successful completion of this course; student shall be able to:

COs	Course Outcomes
B20EJ0504.1	Describe the concepts of WWW including browser and HTTP protocol and summarize the various HTML tags and use them to develop the user- friendly web pages.
B20EJ0504.2	Describe the concepts of WWW including browser and HTTP protocol and summarize the various HTML tags and use them to develop the user- friendly web pages.
B20EJ0504.3	Develop the modern web pages using the HTML and CSS features with different layouts as per need of applications.
B20EJ0504.4	Apply Java Script to develop the dynamic web pages and use server side Scripting with PHP to generate the web pages dynamically using the database connectivity.
B20EJ0504.5	Examine the HTML. Know how to use Dynamic HTML.
B20EJ0504.6	Familiarize server-side scripting language like Perl & PHP.

Conduction of Practical Examination:

1. All laboratory experiments (No. 1 to No. 10) are included for the syllabus of practicalexamination.
2. Students are allowed to pick one experiment from the lot.
3. Strictly follow the instructions as printed on the cover page of answer script.
4. Marks distribution:

Procedure + Conduction + Viva: 08 + 35 +07 = 50 Marks

Change of experiment is allowed only once and marks allotted to the procedure part tobe made zero.

CO-PO-PSO MAPPING

Course	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12	PSO1	PSO2	PSO3
B20EJ0504.1		2	1				1				1		2	2	
B20EJ0504.2		2	3	1	1									3	1
B20EJ0504.3	1	2	1	2				3	1		1			2	1

B20EJ0504.4	1	2	3	1							1	1		2	1
B20EJ0504.5		2	2	3	1									2	2
B20EJ0504.6		1	2	3	1								1	2	
AVG	1	2	2	2	1	0	1	3	1	0	1	1	1	3	1

WEEKWISE EVALUATION OF EACH PROGRAM

ACTIVITY	**MARKS**
Observation +Viva	20
Record	10
TOTAL	30

INTERNAL ASSESSMENT EVALUATION (End of Semester)		
Sl No	**ACTIVITY**	**MARKS**
01	Procedure	7
02	Conduction	8
03	Viva Voce	5
	Total	20

FINAL INTERNAL ASSESSMENT CALCULATION		
Sl. No	**ACTIVITY**	**MARKS**
01	Average of weekly Entries	30
02	Internal Assessment Reduced to	20
	Total	50

Introduction to Web Technology

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 prerequisites are the basic knowledge in HTML, XHTML, CSS, XML, JavaScript, Perl, PHP, Ruby and MySql.

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 uppercase, i.e. there is no difference between<html> and<HTML>
- The order in which parameters of the tagaret given is not significant since each of these parameters is named.

HTML is the standard markup language for Web pages.

With HTML you can create your own Website.

What is HTML?

- HTML stands for Hyper Text Markup Language
- HTML is the standard markup language for creating Web pages
- HTML describes the structure of a Web page
- HTML consists of a series of elements
- HTML elements tell the browser how to display the content
- HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

A Simple HTML Document

Example

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>
<h1>My First Heading</h1>
<p>My first paragraph.</p>
</body>
</html>
```

Example Explained

- The <!DOCTYPE html> declaration defines that this document is an HTML5 document
- The <html> element is the root element of an HTML page
- The <head> element contains meta information about the HTML page
- The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)
- The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
- The <h1> element defines a large heading
- The element defines a paragraph

What is an HTML Element?

An HTML element is defined by a start tag, some content, and an end tag:

<tagname> Content goes here... </tagname>

The HTML **element** is everything from the start tag to the end tag:

<h1>My First Heading</h1>

My first paragraph.

HTML Links

HTML links are defined with the <a> tag:

Example

```
<a href="https://reva.edu.in">This is a link</a>
```

HTML Images

HTML images are defined with the <img> tag.

The source file (src), alternative text (alt), width, and height are provided as attributes:

Example

```
<img src="sample.jpg" alt="reva.edu.in" width="104" height="142">
```

The alt Attribute

The required alt attribute for the <img> tag specifies an alternate text for an image, if the image for some reason cannot be displayed. This can be due to slow connection, or an error in the src attribute, or if the user uses a screen reader.

```
<img src="img_girl.jpg" alt="Girl with a jacket">
```

```
<img src="img_girl.jpg" width="500" height="600">
```

The width and height Attributes

The <img> tag should also contain the width and height attributes, which specifies the width and height of the image (in pixels):

Example

```
<img src="img_girl.jpg" width="500" height="600">
```

HTML is Not Case Sensitive

HTML tags are not case sensitive: <P> means the same as <p>.

The HTML standard does not require lowercase tags, and **demands** lowercase for stricter document types like XHTML.

The
 tag defines a line break, and is an empty element without a closing tag:

Example

```
<p>This is a <br> paragraph with a line break.</p>
```

The style Attribute

The style attribute is used to add styles to an element, such as color, font, size, and more.

<*tagname* style="*property:value;*">

The ***property*** is a CSS property. The ***value*** is a CSS value.

Example

```
<p style="color:red;font-size:30px;font-style:italic">This is a red paragraph.</p>
```

Background Color

The CSS background-color property defines the background color for an HTML element.

Example

Set the background color for a page to powderblue:

```
<body style="background-color:powderblue;">

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>

<body>

<h1 style="background-color:powderblue;">This is a heading</h1>
<p style="background-color:tomato;">This is a paragraph.</p>

</body>
```

Text Color

The CSS color property defines the text color for an HTML element:

Example

```
<h1 style="color:blue;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>
```

Fonts

The CSS font-family property defines the font to be used for an HTML element:

Example

```
<h1 style="font-family:verdana;">This is a heading</h1>
<p style="font-family:courier;">This is a paragraph.</p>
```

Text Size

The CSS font-size property defines the text size for an HTML element:

Example

```
<h1 style="font-size:300%;">This is a heading</h1>
<p style="font-size:160%;">This is a paragraph.</p>
```

Text Alignment

The CSS text-align property defines the horizontal text alignment for an HTML element:

Example

```
<h1 style="text-align:center;">Centered Heading</h1>
<p style="text-align:center;">Centered paragraph.</p>
```

HTML Formatting Elements

Formatting elements were designed to display special types of text:

- <b> - Bold text
- - Important text
- <i> - Italic text
- - Emphasized text
- <mark> - Marked text
- <small> - Smaller text
- <del> - Deleted text
- <ins> - Inserted text
- <sub> - Subscript text
- <sup> - Superscript text

HTML Comment Tag

You can add comments to your HTML source by using the following syntax:

```
<!-- Write your comments here -->
```

HTML Horizontal Rules

The <hr> tag defines a thematic break in an HTML page, and is most often displayed as a horizontal rule.

The <hr> element is used to separate content (or define a change) in an HTML page:

Example

```
<h1>This is heading 1</h1>
<p>This is some text.</p>
<hr>
<h2>This is heading 2</h2>
<p>This is some other text.</p>
<hr>
```

Border Color

You can set the color of borders:

Hello World

Hello World

Hello World

Example

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
<h1 style="border:2px solid Violet;">Hello World</h1>
```

What is CSS?

Cascading Style Sheets (CSS) is used to format the layout of a webpage.

With CSS, you can control the color, font, the size of text, the spacing between elements, how elements are positioned and laid out, what background images or background colors are to be used, different displays for different devices and screen sizes, and much more!

Using CSS

CSS can be added to HTML documents in 3 ways:

- **Inline** - by using the style attribute inside HTML elements
- **Internal** - by using a <style> element in the <head> section
- **External** - by using a <link> element to link to an external CSS file

Inline CSS

An inline CSS is used to apply a unique style to a single HTML element.

An inline CSS uses the style attribute of an HTML element.

The following example sets the text color of the <h1> element to blue, and the text color of the <p> element to red:

Example

```
<h1 style="color:blue;">A Blue Heading</h1>

<p style="color:red;">A red paragraph.</p>
```

Internal CSS

An internal CSS is used to define a style for a single HTML page.

An internal CSS is defined in the <head> section of an HTML page, within a <style> element.

The following example sets the text color of ALL the <h1> elements (on that page) to blue, and the text color of ALL the <p> elements to red. In addition, the page will be displayed with a "powderblue" background color:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1   {color: blue;}
p    {color: red;}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

External CSS

An external style sheet is used to define the style for many HTML pages.

To use an external style sheet, add a link to it in the <head> section of each HTML page:

Example

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

The external style sheet can be written in any text editor. The file must not contain any HTML code, and must be saved with a .css extension.

Here is what the "styles.css" file looks like:

"styles.css":

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color: blue;
  font-family: verdana;
  font-size: 300%;
}
p {
```

```
  color: red;
  font-family: courier;
  font-size: 160%;
}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

CSS Border

The CSS border property defines a border around an HTML element.

Tip: You can define a border for nearly all HTML elements.

Example

Use of CSS border property:

```
p {
  border: 2px solid powderblue;
}
```

CSS Padding

The CSS padding property defines a padding (space) between the text and the border.

Example

Use of CSS border and padding properties:

```
p {
  border: 2px solid powderblue;
  padding: 30px;
}
```

CSS Margin

The CSS margin property defines a margin (space) outside the border.

Example

Use of CSS border and margin properties:

```
p {
  border: 2px solid powderblue;
  margin: 50px;
}
```

Define an HTML Table

```
<!DOCTYPE html>
<html>

<style>
table, th, td {
  border:1px solid black;
}
</style>
<body>

<h2>A basic HTML table</h2>

<table style="width:100%">
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>

<p>To undestand the example better, we have added borders to the table.</p>
</body>
</html>
```

To undestand the example better, we have added borders to the table.

</body>

</html>

Table Cells

Each table cell is defined by a <td> and a </td> tag.

td stands for table data.

Everything between <td> and </td> are the content of the table cell.

Example

```
<!DOCTYPE html>
<html>
<style>
table, th, td {
  border:1px solid black;
}
</style>
<body>
<h2>TD elements define table cells</h2>
<table style="width:100%">
  <tr>
    <td>Emil</td>
    <td>Tobias</td>
    <td>Linus</td>
  </tr>
</table>
<p>To undestand the example better, we have added borders to the table.</p>
</body>
```

```
</html>
```

Collapsed Table Borders

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
</style>
</head>
<body>
<h2>Collapsed Borders</h2>
<p>If you want the borders to collapse into one border, add the CSS border-collapse property.</p>
<table style="width:100%">
  <tr>
    <th>Firstname</th>
    <th>Lastname</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>50</td>
  </tr>
  <tr>
```

```
    <td>Eve</td>
    <td>Jackson</td>
    <td>94</td>
  </tr>
  <tr>
    <td>John</td>
    <td>Doe</td>
    <td>80</td>
  </tr>
</table>
</body>
</html>
```

HTML Table – Colspan

Cell that spans two columns

To make a cell span more than one column, use the colspan attribute.

Name		Age
Jill	Smith	43
Eve	Jackson	57

To make a cell span over multiple columns, use the colspan attribute:

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
```

```
}
</style>
</head>
<body>

<h2>Cell that spans two columns</h2>
<p>To make a cell span more than one column, use the colspan attribute.</p>
<table style="width:100%">
  <tr>
    <th colspan="2">Name</th>
    <th>Age</th>
  </tr>
  <tr>
    <td>Jill</td>
    <td>Smith</td>
    <td>43</td>
  </tr>
  <tr>
    <td>Eve</td>
    <td>Jackson</td>
    <td>57</td>
  </tr>
</table>
</body>
</html>
```

HTML Table – Rowspan

Cell that spans two rows

To make a cell span more than one row, use the rowspan attribute.

Name	Jill
Phone	555-1234
	555-8745

To make a cell span over multiple rows, use the rowspan attribute:

```
<!DOCTYPE html>
<html>
<head>
<style>
table, th, td {
  border: 1px solid black;
  border-collapse: collapse;
}
</style>
</head>
<body>
<h2>Cell that spans two rows</h2>
<p>To make a cell span more than one row, use the rowspan attribute.</p>
<table style="width:100%">
  <tr>
    <th>Name</th>
    <td>Jill</td>
  </tr>
  <tr>
```

```
    <th rowspan="2">Phone</th>
    <td>555-1234</td>
  </tr>
  <tr>
    <td>555-8745</td>
  </tr>
</table>
</body>
</html>
```

HTML Lists

HTML lists allow web developers to group a set of related items in lists.

Example

An unordered HTML list:

- Item
- Item
- Item
- Item

An ordered HTML list:

1. First item
2. Second item
3. Third item
4. Fourth item

```
<!DOCTYPE html>
<html>
<body>
<h2>An Unordered HTML List</h2>
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
```

```
</ul>

<h2>An Ordered HTML List</h2>

<ol>

  <li>Coffee</li>

  <li>Tea</li>

  <li>Milk</li>

</ol>

</body>

</html>
```

HTML Essentials

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

```
<Html>
        <Body>
            <Head>
                    <Title> New Page </title>

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

Basic HTML Tags:-

To create a Check box

```
<input type=checkbox name=C1value=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

```
 <text area rows=2name=S1cols=20></text area>
```

To create a Drop down Menu

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

To create a Hyper Link

<ahref=http://localhost:8080/a.htm>BACK

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. Although they sound and look cool, it is preferred to avoid using too much of theses since they can be tiring and confusing to the users eye especially if the scroll speed is set too high)*

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

To give Background color

<body bg color=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 a HTML Book)

More Miscellaneous Tags:-

1) <h#>… ... </h#>-where **'#'** is a numberrangingfrom1-6. 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 there by preserving the white spaces as well.
3)
-Inserts a "New line" character (similar to'\n').
4) To Draw a Horizontal Line (Horizontal Ruler):

   <hr size=4width="50 %">

5) < b >-**Bold**,< I>-Italics
6) Tables:

```
<Table>
<Caption>Your Caption here </caption> [Optional Tag]

<tr
</tr>
<tr>
<td></td>
</tr>
</table>
```

<th> *Row1, Col1* </th>[th implies Table Header]
<th> *Row2, Col2*</th>

<Td>*Table Definition here*</td>

7).Comments:

<!--*Your Comments here*-->

8).Back ground Images:

<body background="*path name*/abc.gif">

……
</body>

Basics of PHP-

PHP stands for Hypertext Preprocessor.

Essential Features of Perl: -

- It is a server-side scripting language
- It is used for form handling and database access
- PHP supports both procedure a land object-oriented programming
- Syntax and semantics of PHP is closely related to syntax and semantics of Perl and JavaScript
- Variables in PHP don't have to be declared but can be used.

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, and 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 reva ; mysql> show databas

es

;mysql

>use

reva;

Mysql>create table employee (name var char(25),age int);

Mysql>insert into

employee values (“e1”,

21); mysql>insert into

employee values (“e2”,

22);

Mysql>exit;

#..... (The Mysql server is now started and a database along with a table called “employee” is ready for use).

Apache Http Server:

The web server we are using here is Apache Http Server. It is freely downloadable

from the site www.apache.org.Onceyou 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/.Theport number80is*optional*in the URL.

If you have successfully installed the server then you will get a screen shown below:

Steps to Execute PHP Program

- Open notepad and type the XML program and save the file with **.php** extension in var/www/html folder.
- Open the Mozilla Web browser and type the URL “http://localhost/filename.php”
- If you create html file as an interface then store the html file in var/www/html folder and access the URL “http://local host/filename.html”
- The output of the program will be displayed.
- Errors in the php file can be known by compiling as **php file name .php** in terminal

Program No: 1

In today's digital world, information dissemination through printed documents consume lot of time. To overcome this drawback it is better to adopt digital technology for information dissemination, like e-journals, e-books, e-advertisements, etc. Information dissemination through Internet in the form of web content is essential and convenient option. Design and develop static web pages for an online Book store. The pages should resemble like www.amazon.com The website should consist of. Home page, Registration & Login, User profile page, Books catalog, Shopping cart, Payment by credit card, and order confirmation.

Main.html

```
<html>
<head>
   <title>Main Page</title>
   <style type="text/css">
      a
      {
         color:Black;
         font-size:large;
      }
   </style>
</head>
<body style="background-image: url(BgImg2.jpg); background-repeat: no-repeat;
background-size: 100%;">
   <center>
      <h1 style="color:White;">
         Main Page</h1>
      <br />
      <p>
         <b><a href="Home.html">Home</a></b>
      </p>
      <p>
         <b><a href="SignUp.html">Register</a></b>
      </p>
      <p>
         <b><a href="Order.html">Order</a></b>
      </p>
   </center>
</body>
</html>
```

Home.html

```
<html>
<head>
```

```
   <title>Home</title>
</head>
<body  style="background-image: url(cbgimg1.jpg); background-repeat: no-repeat;
background-size: 100%; color:White;">
<center>
   <h1>Welcome to e-Books</h1>
</center>
  <p>Select your book</p>
   <hr />
<center>
<p>
<a href="CheckOut.html"><img src="BgImg1.jpg"  height="250" width="250" /></a>
<a href="CheckOut.html"><img src="BgImg1.jpg"  height="250" width="250" /></a>
<a href="CheckOut.html"><img src="BgImg1.jpg"  height="250" width="250" /></a>
<a href="CheckOut.html"><img src="BgImg1.jpg"  height="250" width="250" /></a>
<a href="CheckOut.html"><img src="BgImg1.jpg"  height="250" width="250" /></a>
</p>
</center>
<center><a href="MainPage.html">Go to Main page</a></center>
</body>
</html>
```

Checkout.html

```
<html>
<head>
   <title>Checkout</title>
</head>
<body style="background-image: url(cbgimg1.jpg); background-repeat: no-repeat;
background-size: 100%; color:White;">
   <center><h1>Checkout</h1></center>
   <p>Enter Card details</p>
   <hr />
   Card No : <input type="text" /> <br /><br />
   Name on Card : <input type="text" /> <br /><br />
   Expiry date : <select >
      <option>2018</option>
      <option>2019</option>
      <option>2020</option>
      <option>2021</option>
   </select> <br /><br />
   CVV No : <input type="text" /> <br /><br />
   Amount paid : <input type="text" /> <br /><br />
   <input type="submit" value="Submit" /><input type="reset" value="reset" /> <br />
   <center>
```

```
   <a href="MainPage.html">Go to Main Page</a>
   </center>
</body>
</html>
```

Order.html

```
<html>
<head>
   <title>Order</title>
</head>
<body  style="background-image: url(cbgimg1.jpg); background-repeat: no-repeat;
background-size: 100%; color:White;">
   <center>
      <h1>
         Order</h1>
   <hr />
      <p>
         Your Order</p>
      <a href="MainPage.html">Go to Main Page</a></center>
</body>
</html>
```

Signup.html

```
<html>
<head>
   <title>Sign up</title>
</head>
<body style="background-image: url(cbgimg1.jpg); background-repeat: no-repeat;
background-size: 100%; color:White;">
<center><h1>Sign up</h1></center>
   <p>Enter your details</p>
   <hr />
   First Name : <input type="text" /> <br /><br />
   Last Name : <input type="text" /> <br /><br />
   Email Id   : <input type="text" /> <br /><br />
   User Id : <input type="text" /> <br /><br />
   Password : <input type="password" /> <br /><br />
   Phone No : <input type="text" /> <br /><
   Day : <input type="text" /> <br /><br />
   <input type="submit" value="Submit" /><input type="reset" value="reset" /> <br />
   <center>
   <a href="MainPage.html">Go to Main Page</a>
   </center>
</body>
```

</html>

OUTPUT:

Main.html

Home

Select a book

Register


Sign up
Enter your details
First Name :
Last Name :
Email Id :
User Id :
Password :
Phone No :
Day :
Submit
reset
Go to Main Page

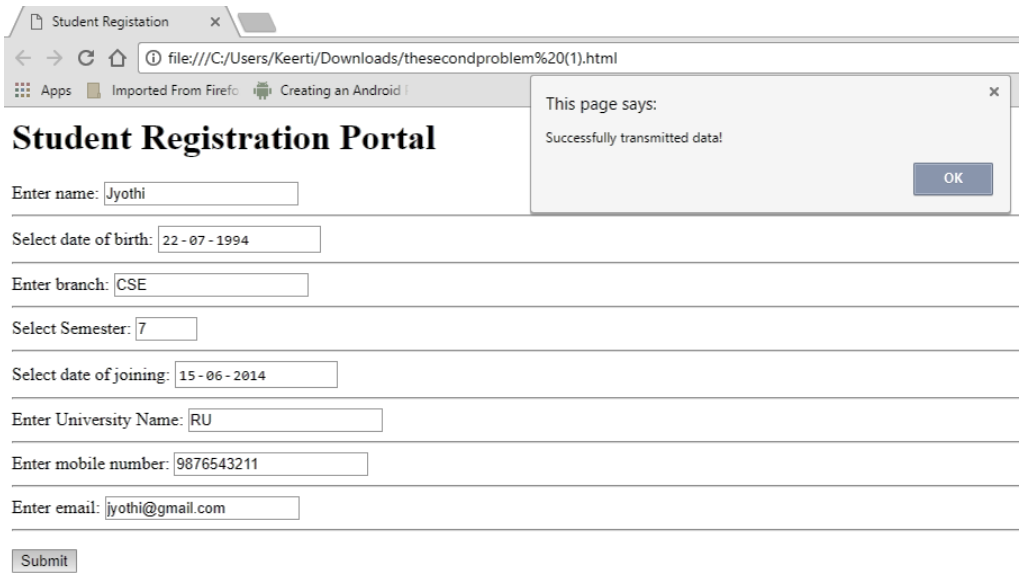
Program No: 2

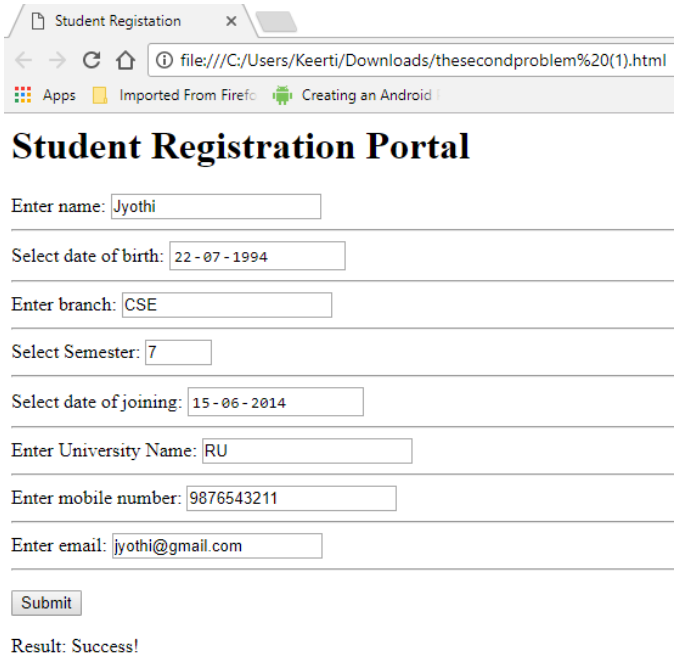
Internet or online services works on clients and server model. A client is a web browser through which users make requests, which contain input required, for service from the server to perform tasks. Server is a program running on a dedicated computer. Performance of any service or server depends on its throughput. Server throughput deteriorates when users send more and more invalid requests for service and thus results in wastage of server resources that are very precious. As a solution to this problem design a web page that takes student details such as Name, branch, Semester, University, date of admission, mobile number, email id and check for validity or correctness of the input data by writing a JavaScript to validate these fields.

2.html

```
<html>
<head>
<title>Student Registration</title>
</head>
<body>
	<div align="Left">
	<h1>Student Registration Portal</h1>
	<form id="xyz">
		<label for="name">Enter name: </label>
		<input type="text" id="name" /><br /><hr />
		<label for="dob">Select date of birth: </label>
		<input type="date" id="dob" /><br /><hr />
		<label for="branch">Enter branch: </label>
		<input type="text" id="branch" /><br /><hr />
		<label for="semester">Select Semester: </label>
		<input type="number" id="semester" max="8" min="0" /><br /><hr />
		<label for="doj">Select date of joining: </label>
		<input type="date" id="doj" /><br /><hr />
		<label for="university">Enter University Name: </label>
		<input type="text" id="university" /><br /><hr />
		<label for="mobile">Enter mobile number: </label>
		<input type="text" id="mobile" /><br /><hr />
		<label for="email_add">Enter email: </label>
		<input type="email" id="email_add" /><br /><hr />
	</form>
	<button onclick="validate()">Submit</button>
	<p>
		Result: <span id="result"></span>
	</p>
	</div>
</body>
<script>
function validate()
```

```
{
        var result_text = document.getElementById("result");
        var dob = document.getElementById("dob").value;
        var birth_year = parseInt(dob.substring(0, 4));
        var doj = document.getElementById("doj").value;
        var join_year = parseInt(doj.substring(0,4));
        if (join_year - birth_year < 17)
        {
                result_text.innerHTML = "Too young to have started college!"

                return;
        }
        var branch = document.getElementById("branch").value;
        if (branch.search(/(CSE|ECE|ME|CE|EEE|BCA|MCA)/i) == -1)
        {
                result_text.innerHTML = "Invalid branch!";
                return;
        }
        var mobile_no = document.getElementById("mobile").value;
        if (mobile_no.search(/^[0-9]+$/) == -1 || mobile_no.length != 10)
        {
                result_text.innerHTML = "Invalid mobile number!";
                return;
        }
        var email = document.getElementById("email_add").value;
        if (email.search(/^(([^<>()\[\]\\.,;:\s@"]+(\.[^<>()\[\]\\.,;:\s@"]+)*)|(".+"))@((\[[0-
9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}])|(([a-zA-Z\-0-9]+\.)+[a-zA-Z]{2,}))$/) == -1)
        {
                result_text.innerHTML = "Invalid email ID";
                return;
        }
        alert('Successfully transmitted data!');
        result_text.innerHTML = "Success!";
}
</script>
</html>
```


Student Registation
file:///C:/Users/Keerti/Downloads/thesecondproblem%20(1).html
Apps
Imported From Firefo
Creating an Android
This page says:
Successfully transmitted data!
OK
Student Registration Portal
Enter name: Jyothi
Select date of birth: 22-07-1994
Enter branch: CSE
Select Semester: 7
Select date of joining: 15-06-2014
Enter University Name: RU
Enter mobile number: 9876543211
Enter email: jyothi@gmail.com
Submit


Student Registation
file:///C:/Users/Keerti/Downloads/thesecondproblem%20(1).html
Apps
Imported From Firefo
Creating an Android
Student Registration Portal
Enter name: Jyothi
Select date of birth: 22-07-1994
Enter branch: CSE
Select Semester: 7
Select date of joining: 15-06-2014
Enter University Name: RU
Enter mobile number: 9876543211
Enter email: jyothi@gmail.com
Submit
Result: Success!

Program No: 3

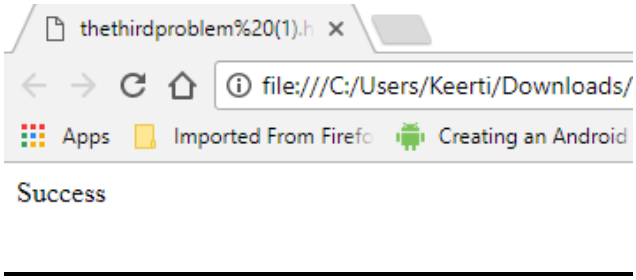
Clients interact with servers by sending service requests that contain input required to complete the requested task or service. Input required for requested service may be collected through a web page, that acts as an interface between users and the server, in the form of text fields, text areas, radio buttons, push buttons and so on. Hence it is better to instruct or help clients to input correct data through web page by displaying appropriate error messages or alerts as and when users supply wrong input using event handlers. To demonstrate this task, design and develop a web page using JavaScript, XHTML that collects the SRN (Valid format is: Any letter followed by two digits, followed by two letters then followed by three digits). Include event handler for the form elements that collects information to validate the input. Messages must be produced in the alert windows as and when errors are detected.

3.html

```
<html>
<head>
<title>Enter the Details</title>
<script>
	function validate()
	{
		var srn = document.getElementById("srn_check").value;
		if (srn.search(/^[A-Z a-z]+[0-9]+[0-9]+[A-Z a-z]+[A-Z a-z]+[0-9]+[0-9]+[0-9]+$/) == -1 || srn.length != 8)
		{
			document.write("Invalid SRN.....!!!!!Enter the SRN properly");
			return;
		}
		document.write("Success");
	}
</script>
</head>
<body>
	<h1>Please enter the following field</h1>
	<form>
		<labeL>Enter SRN: </label>
		<input type="text" id="srn_check" /><br /><hr />
		<input type="submit" value="Check" onclick="validate()" />
	</form>
</body>
</html>
```

OUTPUT:


Enter the Details
file:///C:/Users/Keerti/Downloads/thethirdproblem%20(1).html
Apps
Imported From Firefo
Creating an Android
Please enter the following field
Enter SRN: R14cs001
Check


thethirdproblem%20(1).h
file:///C:/Users/Keerti/Downloads/
Apps
Imported From Firefo
Creating an Android
Success

Program No: 4

Dynamic web content is the information that is retrieved from one or more web servers depending upon what information client have requested for, and composed in response to users' requests. Advanced web technologies play a vital role in storage, processing and retrieval of dynamic web content from web servers. Hence it is important to use advanced web technologies such as XML to improve the efficiency in data retrieval. Create and save XML document for students information and display the same using cascaded style sheet.

4.xml

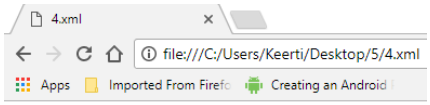
```
<?xml version="1.0" encoding="ISO-8859-1"?>
<?xml-stylesheet type="text/css" href="4.css"?>
<CATALOG>
<STUDENT>
<NAME>Anurag</NAME>
<SRN>R14CS001</SRN>
<BRANCH>CSE</BRANCH>
<SEMESTER>3</SEMESTER>
<UNIVERSITY>REVA</UNIVERSITY>
<COUNTRY >India</COUNTRY>
</STUDENT>
<STUDENT>
<NAME>Arjun</NAME>
<SRN>R14CS021</SRN>
<BRANCH>CSE</BRANCH>
<SEMESTER>5</SEMESTER>
<UNIVERSITY> REVA </UNIVERSITY>
<COUNTRY> India </COUNTRY>
</STUDENT>
<STUDENT>
<NAME>William</NAME>
<SRN>R14CS099</SRN>
<BRANCH>CSE</BRANCH>
<SEMESTER>7</SEMESTER>
<UNIVERSITY> REVA </UNIVERSITY>
<COUNTRY> India </COUNTRY>
</STUDENT>
</CATALOG>
```

4.css

```
CATALOG
{
background-color: #ffffff;
width: 100%;
}
STUDENT
```

```
{
display: block;
margin-bottom: 30pt;
margin-left: 0;
}
NAME
{
color: #FF0000;
font-size: 20pt;
}
SRN
{
color: #0000FF;
font-size: 20pt;
}
BRANCH,SEMESTER,UNIVERSITY,COUNTRY
{
display: block;
color: #000000;
margin-left: 20pt;
}
```

OUTPUT:



Anurag R14CS001
CSE
3
REVA
India

Arjun R14CS021
CSE
5
REVA
India

William R14CS099
CSE
7
REVA
India

Program No: 5

Information technology has become part and parcel of humanity to such an extent that people can shop anything online, from anywhere, at any time using an electronic device that has access to Internet. This has brought in the concept of virtual stores which provide products at less cost. To improve sales it is mandatory to organize items catalog based on item name, item price, and manufacturer so on. For such online shopping sites, look and feel is an obvious requirement which can be achieved using CSS & XSLT. Design a document using CSS and XSLT to create a catalog of items for an online electronic shopping.

5a.xml

```
<?xml version = "1.0"?>
<?xml-stylesheet type="text/css" href="5a.css" ?>
 <RU>
     <b> Books </b><br />
     <e> Electronics </e><br />
     <s> Stationaries </s><br />
     <c> Cloths</c><br />
     <o> other </o><br />
 </RU>
```

5a.css

```
b{ display:block;color:blue;font-style:italic;}
e{ display:block;color:green;font-style:italic;}
s{ display:block;color:red;font-style:italic;}
c{ display:block;color:blue;font-style:italic;}
o{ display:block;color:black;font-style:italic;}
```

5b.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="5b.xsl" ?>
<RU>
     <b> Books </b>
     <e> Electronics </e>
     <s> Stationaries </s>
     <c> Cloths </c>
     <o> Other </o>
 </RU>
```

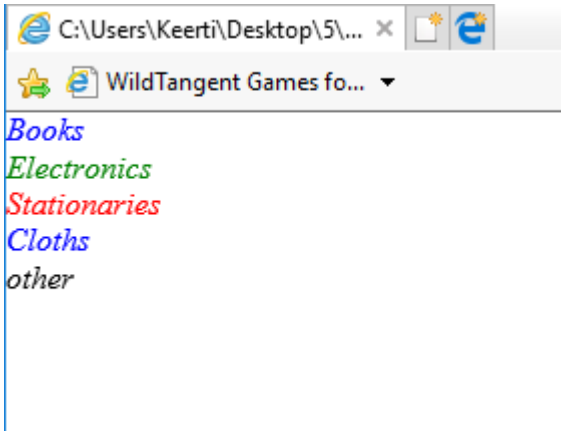
5b.xsl

```
<?xml version = "1.0"?> <xsl:stylesheet version = "1.0" xmlns:xsl =
"http://www.w3.org/1999/XSL/Transform" xmlns="http://www.w3.org/1999/xhtml">
<xsl:template match = "RU">
 <html><head><title> Style sheet for 5b.xml </title>
 </head><body>
 <h2> Shopping </h2>
```

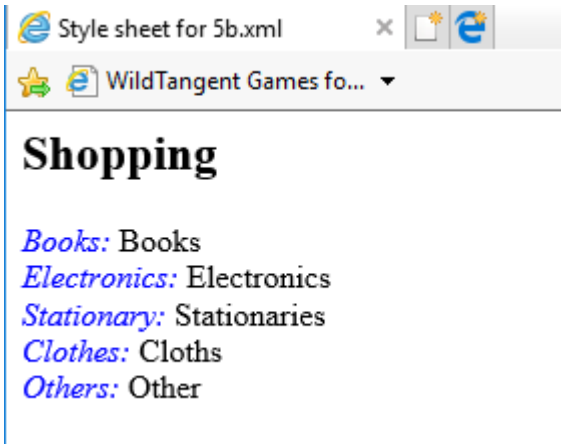
```
<span style = "font-style: italic; color: blue;"> Books:
</span> <xsl:value-of select = "b" /> <br />

<span style = "font-style: italic; color: blue;"> Electronics:
</span> <xsl:value-of select = "e" /> <br /

<span style = "font-style: italic; color: blue;"> Stationary:
</span> <xsl:value-of select = "s" /> <br />

<span style = "font-style: italic; color: blue;"> Clothes:
</span> <xsl:value-of select = "c" /> <br />

<span style = "font-style: italic; color: blue;"> Others:
</span> <xsl:value-of select = "o" /> <br />

</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

OUTPUT:

Open 5a.xml in browser



Open 5b.xml in browser



Program No: 6

In any business organization, employees keep traveling across different geographical locations and at the same time they want to be connected to their organization's computing resources such as email server, database server, file server, etc. to retrieve information such as sales details, assigning tasks to employees, and upload inspection site details, so on. Using PHP develop a web page that accepts book information such as ISBN number, title, authors, edition and publisher and store information submitted through web page in MySQL database. Design another web page to search for a book based on book title specified by the user and displays the search results with proper headings.

book.html

```
<html>
<body>
        <form action="http://localhost/cgi-bin/bookinsert.php"  method="post">
        Title:          <input type="text"name="title"/>
        Author:                 <input type="text" name="author"/>
        Publisher:      <input type="text" name="publisher"/>
                        <input type="submit"/>
        </form>
</body>
</html>
```

bookinsert.php

```
<html>
<body>
        <?php
                $con = mysql_connect("localhost","root","") or die(mysql_error());
                mysql_select_db ("test") or die(mysql_error());
                $sql="insert into books (title, author, publisher) values ('$_POST[title]', r]',
                '$_POST[publisher]')";

                if (!mysql_query($sql,$con))
                {
                        die('Error: ' . mysql_error());
                }

        echo "1 record added";
        mysql_close($con);
        ?>
        <form action="bookresult.php" method="post">
                        Title:  <input type="text" name="title" />
                                <input type="submit" />
        </form>
</body></html>
```

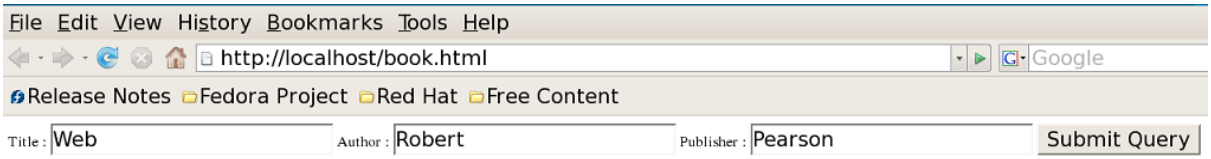
bookresult.php

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

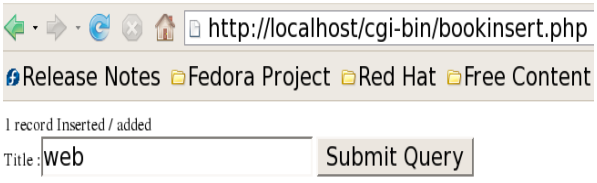
```
<?php
        $con = mysql_connect("localhost","root","") or die(mysql_error());
        mysql_select_db("test") or die(mysql_error());
        $result = mysql_query("select * from books where title= '$_POST[title]'");
        if(!$result)
        {
                echo "There is no record";
        }
    echo "<table border='1'><tr><th>Title</th><th>Author</th>th>Publisher</th></tr>";
    while($row = mysql_fetch_array($result))
    {
                echo "<tr>";
                echo "<td>" . $row['title'] . "</td>";
                echo "<td>" . $row['author'] . "</td>";
                echo "<td>" . $row['publisher'] . "</td>";
                echo "</tr>";
    }
    echo "</table>";
    mysql_close($con);
    ?>
</body>
</html>
```

OUTPUT:

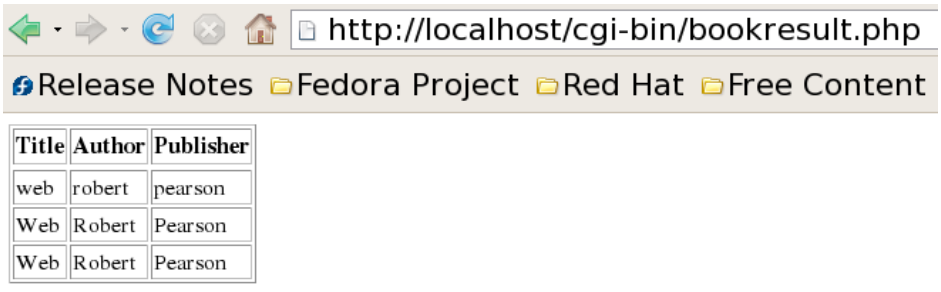
Book.html



Bookinsert.php



Bookresult.php



Program No: 7

Using computers without graphical user interfaces require the knowledge about syntax of computer commands and programming languages, also this makes users to feel that the use of computers is difficult and cumbersome. This impression of users on computers can be changed by providing good and easy-to-use graphical user interfaces which play vital role in use of computer applications or software without worrying about syntax of programming languages or computer commands. In fact computer software with good and easy-to-use graphical user interfaces will have large number of users. a) Design HTML page that takes UNIX command as input in a text field and submit it to a Perl program that executes given command and display the output on the web page b) Write a Perl program to keep track of the number of visitors to a web page and display the count of visitors with proper headings.

7.html

```
<html>
<head><title>Unix Command</title></head>
<body>
        <form method=GET action="http://localhost/cgi-bin/7a.pl" >
        <center><br><h1>ENTER UNIX COMMAND TO BE EXECUTED</h1>
        <input type="text" name="msg">
        <input type="submit" value="CLICKME">
        </form>
</body>
</html>
```

7a.pl

```
#!/usr/bin/perl
use CGI':standard';
        print "con tent-type:text/html\n\n";
        $c=param('msg');
        system($c);
exit(0);
```

7b.pl

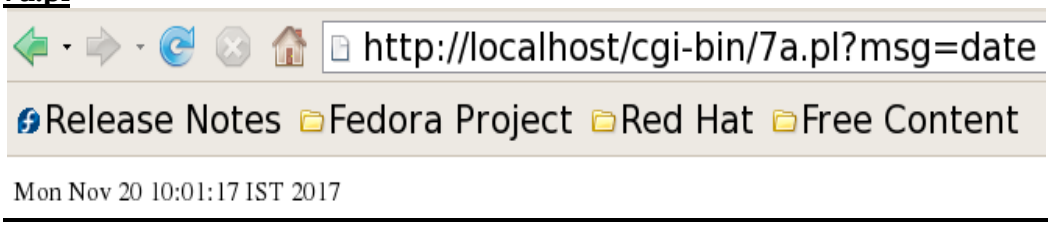
```
#!/usr/bin/perl
print "content-type:text/html ";
        $count_file="/var/www/cgi-bin/ru.txt";
        if(open(FILE,'<'.$count_file))
        {
                $no_accesses=<FILE>;
                close(FILE);
                if(open(FILE,'>'.$count_file))
                {
                        $no_accesses++;
                        print FILE $no_accesses;
                        close(FILE);
                        print "no. of visitors:",$no_accesses;
                }
                else
                {
                        print "[can't write to data file]";
```

```
                }
        }
        else
        {
                print "[sorry]";
        }
exit(0);
```

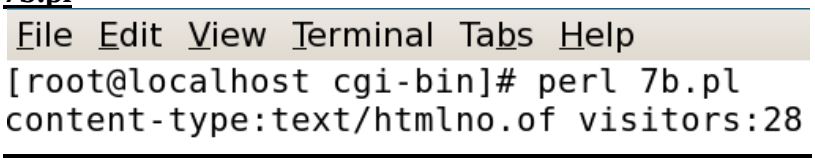
OUTPUT:

7.html

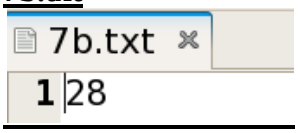
7a.pl



7b.pl



7b.txt



Program No: 8

Databases are the storage systems used by most of the business and information technology enterprises as back end. When users generate data using GUI, for ex. personal information, data are sent to back end database for storage and also users can retrieve this data as and when required from the back end (database) to the front GUI. In the real world there are several databases such as Oracle, DB2, MySQL, SQL Server, MS-Access, DBMongo, etc. To illustrate the process of generating data from the front end and store it on back end database then retrieve the available data from the back end database, write a Perl program to read personal information of a person such as first name, last name, age, permanent address and pin code entered by the user into a table created in MySQL. Read the same information from the database and display on the front end.

First Name	Last Name	Age	Address	Pincode
Ram	Kumar	21	REVA University	560064
Anil	Vinay	30	REVA University	560064

8.html

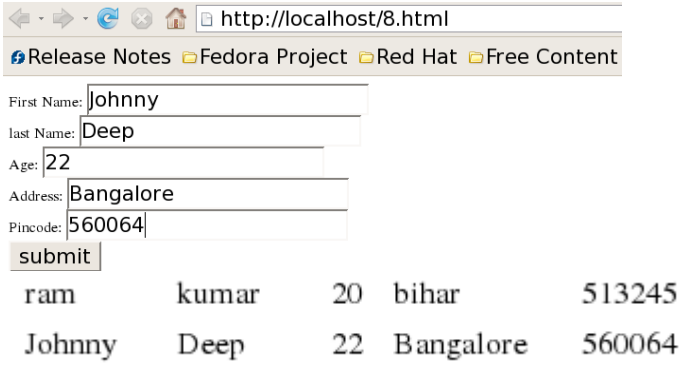
```
<html>
<body>
        <form action="http://localhost/cgi-bin/8.pl" method="get">
          First Name :  <input type="text" name="fname"> <br>
          Last Name :  <input type="text" name="lname"> <br>
          Age :  <input type="text" name="age"> <br>
          Address :  <input type="text" name="address"> <br>
          Pincode :  <input type="text" name="pincode"> <br>
                     <input type="submit" value="submit">
        </form>
</body>
</html>
```

8.pl

```
#!/usr/bin/perl
print "content-type:text/html ";
        print "<html><title>Result of the insert operation </title>";
        use CGI ':standard';
        use DBI;
        $dbh=DBI->connect("DBI:mysql:test","root","");
        $fname=param("fname");
        $lname=param("lname");
        $age=param("age");
        $address=param("address");
        $pincode=param("pincode");
        $qh=$dbh->prepare("insert into stud
values('$fname','$lname','$age','$address','$pincode')");
        $qh->execute();
        $qh=$dbh->prepare("select * from stud");
        $qh->execute();
        print "<table border
size=1><tr><th>Firstname</th><th>LastName</th><th>Age</th<th>Address</th><th>Pinc
ode</th></tr>";
while ( ($fname,$lname,$age,$address,$pincode) 56  >fetchrow())
{
```

```
      print
"<tr><td>$fname</td><td>$lname</td><td>$age</td><td>$address</td><td>$pincode</td>
</tr>";
}
print "</table>";
$qh->finish();
$dbh->disconnect();
print"</HTML>";
```

OUTPUT:



Program No: 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.

9.html

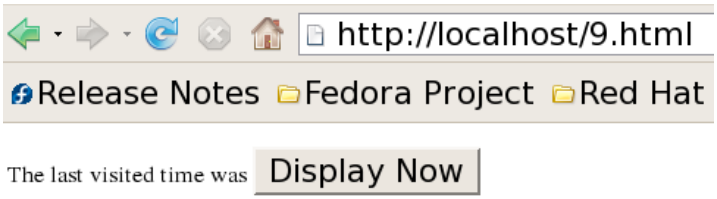
```
<html >
<head> <title>Cookies</title> </head>
<body>
<form action="http://localhost/cgi-bin/9.php" method="post">
<p> The last visited time was <input type="submit" value="Display Now"/> </p>
</form>
</body>
</html>
```

9.php

```
<?php
$present_time=date("H:i:s-m/d/y");
$expiry= 60 * 60 *24 *60 + time();
setcookie("Lastvisit",$present_time, $expiry);
if(isset($_COOKIE["Lastvisit"]))
{
echo "Cookie has been set";
echo "The current time of the system is";
echo $present_time;
echo "The Last visited Time and Date";
echo $_COOKIE["Lastvisit"];
}
else
echo "You've got some old cookies!";
?>
```

OUTPUT:

9.html



Program No: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.

10.html

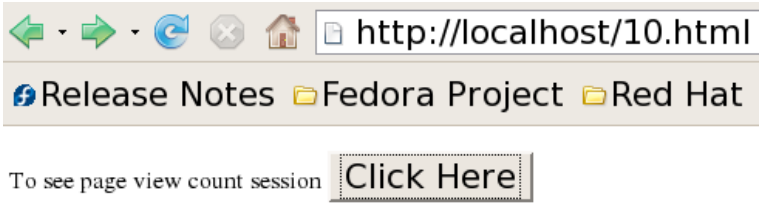
```
<html>
<head> <title>SESSION PROGRAM </title> </head>
<body>
<form action="http://localhost/cgi-bin/10.php" method="post">
<p> To see page views count in session <input type="submit" value="Click Here"/> </p>
</form>
</body>
</html>
```

10.php

```
<?php
session_start();
if (!isset($_SESSION))
{
$_SESSION["count"] = 1;
echo "<p>Counter initialized</p>\n";
}
else { $_SESSION["count"]++; }
echo "<p>This page has been viewed <b>$_SESSION[count]</b> times.</p>".
"<p>reload this page to increment</p>";
?>
```

OUTPUT:

10.html



ADDITIONAL QUESTIONS AND ANSWERS

Q1 What is the meaning of Webtech?

Ans Webtech stands for Web Technologies, and it refers to the various technologies used for developing websites. Webtech includes a wide variety of tools, programming languages, and protocols that allow you to create interactive web applications.

Q2 What is the difference between session storage and local storage objects in HTML?

Ans The difference between session storage and local storage is described as below:

Local Storage: In local storage, objects don't have any expiry for the data stored in it and local storage object have no permission to delete the data after the browser is closed.

Session Storage: Whereas session storage keeps objects in it up to when the session is alive and it clears the data stored in it when the session is closed.

Q3 Differentiate between hypertext and hyperlink.

Ans The hypertext is basically a simple text that contains a link which redirects the user to somewhere else on the computer network. While the hyperlink is a link that allows users to navigate between difference computer resources like webpages.

Q4 Differentiate between web browser and web server

Ans A web browser acts as a link/ interface between a client and a server. Its primary function is to display various web documents to the clients requesting them. A web server functions to accept browser requests, generate responses, maintain the web apps, and accept the client data.

Q5 What are the basic text formatting tags used in HTML?

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

Q6 What is HTML?

Ans HTML stands for Hyper Text Markup Language. It is a language of World Wide Web. It is a standard text formatting language which is used to create and display pages on the Web.

Q7 What are Tags?

Ans HTML tags are composed of three things: opening tag, content and ending tag. Some tags are unclosed tags. HTML documents are made of two things: content, and tags content is placed between tags to display data on the web page.

Q8 What is the difference between HTML elements and tags?

Ans HTML elements communicate to the browser to render text. When the elements are surrounded by brackets <>, they form HTML tags. Most of the time, tags come in pair and surround content.

Q9 How to make a picture a background image of a web page?

Ans To make a picture a background image on a web page, you should put the following tag code after the </head> tag.

<body background = "image.gif">

Here, replace the "image.gif" with the name of your image file which you want to display on your web page.

Q10 If I do not put <!DOCTYPE html> will HTML 5 work?

Ans No, browser will not be able to identify that it is a HTML document and HTML 5 tags will not function properly.