

Web Technology and its Applications

Time: 3 hrs.

Max. Marks: 80

Note : Answer any FIVE full questions, selecting ONE full question from each module.

Module - 1

1. a. Briefly explain the history of markup languages. (04 Marks)

Ans. The Internet of the late 1980s and the very early 1990s did not resemble the Internet we know today. During these early years, email and text-based systems were the extent of the Internet experience. This transition from the old terminal and text-only Internet of the 1980s to the Internet of today is of course due to the invention and massive growth of the World Wide Web. This invention is usually attributed to the British Tim Berners-Lee (now Sir Tim Berners-Lee), who, along with the Belgian Robert Cailliau, published a proposal in 1990 for a hypertext system while both were working at CERN in Switzerland. Shortly thereafter Berners-Lee developed the main features of the web.⁴

This early web incorporated the following essential elements that are still the core features of the web today:

1. A Uniform Resource Locator (URL) to uniquely identify a resource on the WWW.
2. The Hypertext Transfer Protocol (HTTP) to describe how requests and responses operate.
3. A software program (later called web server software) that can respond to HTTP requests.
4. Hypertext Markup Language (HTML) to publish documents.
5. A program (later called a browser) that can make HTTP requests from URLs and that can display the HTML it receives.

- b. Write a note on XHTML and HTML 5 (04 Marks)

Ans. **XHTML**

- XHTML stands for EXtensible HyperText Markup Language
- XHTML is almost identical to HTML
- XHTML is stricter than HTML
- XHTML is HTML defined as an XML application
- XHTML is supported by all major browsers

HTML 5

- New parsing rules that are not based on SGML but are oriented towards flexible parsing and compatibility.
- Support of use of inline Scalar Vector Graphics (SVG) and Mathematical Markup Language (MathML) in text/html.
- New available elements include article, aside, audio, bdi, canvas, command, datalist, details, embed, figcaption, figure, footer, header, hgroup, keygen, mark, meter, nav, output, progress, rp, rt, ruby, section, source, summary, time, video

- and wbr.
- New available types of form controls include dates and times, email, url, search, number, range, tel and color.
- New available attributes of charset on meta and async on script.
- Global attributes that can be applied for every element that include id, tabindex, hidden, data-* or customer data attributes.

c. Explain i) <a> ii) iii) <p> iv) <div> elements with examples (08 Marks)

Ans. i) <a>

The <a> tag defines a hyperlink, which is used to link from one page to another. The most important attribute of the <a> element is the href attribute, which indicates the link's destination. By default, links will appear as follows in all browsers:

- An unvisited link is underlined and blue
- A visited link is underlined and purple
- An active link is underlined and red

Example: Visit Google.com!

ii)

The tag defines an image in an HTML page. The tag has two required attributes: src and alt.

Note: Images are not technically inserted into an HTML page, images are linked to HTML pages. The tag creates a holding space for the referenced image.

Tip: To link an image to another document, simply nest the tag inside <a> tags

Example:

iii) <p>

The <p> tag defines a paragraph. Browsers automatically add some space (margin) before and after each <p> element. The margins can be modified with CSS (with the margin properties).

Example: <p>This is some text in a paragraph.</p>

iv) <div>

The <div> tag defines a division or a section in an HTML document.

The <div> element is often used as a container for other HTML elements to style them with CSS or to perform certain tasks with JavaScript.

Example:

```
<div style="background-color:lightblue">
  <h3>This is a heading</h3>
  <p>This is a paragraph.</p>
</div>
```

OR

2. a. With an example explain different levels of style sheets. (08 Marks)

Ans. Three Types of CSS

CSS comes in three types:

1. In a separate file (external)
2. At the top of a web page document (internal)
3. Right next to the text it decorates (inline)

1) External style sheets are separate files full of CSS instructions (with the file extension .css). When any web page includes an external stylesheet, its look and feel will be controlled by this CSS file (unless you decide to override a style using one of these next two types). This is how you change a whole website at once. And that's perfect if you want to keep up with the latest fashion in web pages without rewriting every page!

File.css

```
body {  
    background-color:powderblue;  
}  
.main {  
    text-align:center;  
}  
.GFG {  
    color:#009900;  
    font-size:50px;  
    font-weight:bold;  
}  
#geeks {  
    font-style:bold;  
    font-size:20px;  
}
```

Index.html

```
<!DOCTYPE html>  
<html>  
    <head>  
        <link rel="stylesheet" href="file.css"/>  
    </head>  
    <body>  
        <div class = "main">  
            <div class ="GFG">RamforRam</div>  
            <div id ="geeks">A computer science portal</p>  
        </div>  
    </body>  
</html>
```

2) Internal styles are placed at the top of each web page document, before any of the content is listed. This is the next best thing to external, because they're easy to find, yet allow you to 'override' an external style sheet -- for that special page that wants to be a nonconformist!

```
<!DOCTYPE html>
```

```

<html>
  <head>
    <title>Internal CSS</title>
    <style>
      .main {
        text-align:center;
      }
      .GFG {
        color:#009900;
        font-size:50px;
        font-weight:bold;
      }
      .geeks {
        font-style:bold;
        font-size:20px;
      }
    </style>
  </head>
  <body>
    <div class = "main">
      <div class ="GFG">RamforRam</div>
      <div class ="geeks">A computer science portal <p>
    </div>
  </body>
</html>

```

3) Inline styles are placed right where you need them, next to the text or graphic you wish to decorate. You can insert inline styles anywhere in the middle of your HTML code, giving you real freedom to specify each web page element. On the other hand this can make maintaining web pages a real chore!

```

<!DOCTYPE html>
<html>
  <head>
    <title>Inline CSS</title>
  </head>

  <body>
    <p style = "color:#009900;
      font-size:50px;
      font-style:italic;
      text-align:center;">
      RamforRam</p>
  </body>
</html>

```

b. List the different selectors available in CSS and explain in detail (08 Marks)

Ans. A CSS selector is the part of a CSS rule set that actually selects the content you want to style.

Universal Selector

The universal selector works like a wild card character, selecting all elements on a page. Every HTML page is built on content placed within HTML tags. Each set of tags represents an element on the page. Look at the following CSS example, which uses the universal selector:

```
* {  
    color: green;  
    font-size: 20px;  
    line-height: 25px;  
}
```

The three lines of code inside the curly braces (color, font-size, and line-height) will apply to all elements on the HTML page. As seen here, the universal selector is declared using an asterisk. You can also use the universal selector in combination with other selectors.

Element Type Selector

Also referred to simply as a “type selector,” this selector must match one or more HTML elements of the same name. Thus, a selector of nav would match all HTML nav elements, and a selector of would match all HTML unordered lists, or elements.

The following example uses an element type selector to match all elements:

```
ul {  
    list-style: none;  
    border: solid 1px #ccc;  
}
```

To put this in some context, here’s a section of HTML to which we’ll apply the above CSS:

```
<ul>  
    <li>Fish</li>  
    <li>Apples</li>  
    <li>Cheese</li>  
</ul>  
  
<div class="example">  
    <p>Example paragraph text.</p>  
</div>
```

```
<ul>  
    <li>Water</li>  
    <li>Juice</li>
```

Maple Syrup

There are three main elements making up this part of the page: Two elements and a <div>. The CSS will apply only to the two elements, and not to the <div>. Were we to change the element type selector to use <div> instead of , then styles would apply to the <div> and not to the two elements.

ID Selector

An ID selector is declared using a hash, or pound symbol (#) preceding a string of characters. The string of characters is defined by the developer. This selector matches any HTML element that has an ID attribute with the same value as that of the selector, but minus the hash symbol.

Here's an example:

```
#container {  
    width: 960px;  
    margin: 0 auto;  
}
```

This CSS uses an ID selector to match an HTML element such as:

```
<div id="container"></div>
```

Class Selector

The class selector is the most useful of all CSS selectors. It's declared with a dot preceding a string of one or more characters. Just as is the case with an ID selector, this string of characters is defined by the developer. The class selector also matches all elements on the page that have their class attribute set to the same value as the class, minus the dot.

Take the following rule set:

```
.box {  
    padding: 20px;  
    margin: 10px;  
    width: 240px;  
}
```

These styles will apply to the following HTML element:

```
<div class="box"></div>
```

Module - 2

3. a. Explain different form widgets created with the <input> tag.

(08 Marks)

Ans. <legend>

<label>
<input type="text">
<input type="search">
<input type="url">

```
<input type="email">
<input type="tel">
<input type="password">
<input type="datetime">
<input type="date">
<input type="month">
<input type="week">
<input type="time">
<input type="datetime-local">
<input type="number">
<input type="range">
<input type="color">
<input type="checkbox">
<input type="radio">
<input type="file">
<input type="submit">
<input type="image">
<input type="reset">
<input type="button">
<button>button
<select>
<datalist>
<textarea>
<progress>
<meter>
<video>
```

b. Write HTML Code for the following table:

(08 Marks)

Time Day		9.00 am to 1.15 pm	2.00 pm to 5.00 pm
Mon to Fri	Sub	Theory class	ML/WTA Lab
	FI		AD block, 1 st *floor
Sat.	Sub	Extra curricular activity	
	FI		

Ans. <!DOCTYPE html>

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

```

</head>
<body>
<table style="width:100%">
    <tr>
        <th colspan = "2">TimeDay</th>
        <th>9.00AM to 1.15PM</th>
        <th>2.00Pm to 5.00PM</th>
    </tr>
    <tr>
        <td rowspan = "2">Mon to Fri</td>
        <td>Sub</td>
        <td>Theory Class</td>
        <td>ML/WTA Lab</td>
    </tr>
    <tr>
        <td>FI</td>
        <td>ABC/EFG/XYZ</td>
        <td>AD Block, 1st Floor</td>
    </tr>
    <tr>
        <td rowspan = "2">      </td>
        <td>Sub</td>
        <td colspan = "2" rowspan = "2">Extra Curricular Activity</td>
    </tr>
    <tr>
        <td>FI</td>
    </tr>
</table>
</body>
</html>

```

OR

4. a. Discuss the difference between relative and absolute positioning. (08 Marks)

Ans. Relative Positioning:

- Relative. This type of positioning is probably the most confusing and misuse.
- What it really means is “relative to itself”. If you set position: relative; on element but no other positioning attributes (top, left, bottom or right), it will h

no effect on its positioning at all, it will be exactly as it would be if you left it as position: static;

- But if you do give it some other positioning attribute, say, top: 10px;, it will shift its position 10 pixels down from where it would normally be. I'm sure you can imagine, the ability to shift an element around based on its regular position is pretty useful. I find myself using this to line up form elements many times that have a tendency to not want to line up how I want them to.
- There are two other things that happen when you set position: relative; on an element that you should be aware of. One is that it introduces the ability to use z-index on that element, which doesn't really work with statically positioned elements.
- Even if you don't set a z-index value, this element will now appear on top of any other statically positioned element. You can't fight it by setting a higher z-index value on a statically positioned element. The other thing that happens is it limits the scope of absolutely positioned child elements.
- Any element that is a child of the relatively positioned element can be absolutely positioned within that block

Absolute Positioning

- Absolute. This is a very powerful type of positioning that allows you to literally place any page element exactly where you want it. You use the positioning attributes top, left, bottom, and right to set the location. Remember that these values will be relative to the next parent element with relative (or absolute) positioning.
- If there is no such parent, it will default all the way back up to the <html> element itself meaning it will be placed relative to the page itself.
- The trade-off (and most important thing to remember) about absolute positioning is that these elements are removed from the flow of elements on the page.
- An element with this type of positioning is not affected by other elements and it doesn't affect other elements. This is a serious thing to consider every time you use absolute positioning. Its overuse or improper use can limit the flexibility of your site.

b. What does floating an element do in CSS? How do you float an element?

(08 Marks)

Ans. You can float elements to the left or right, but only applies to the elements that generate boxes that are not absolutely positioned. Any element that follows the floated element will flow around the floated element on the other side.

The float property may have one of the three values:

Value	Description
left	The element floats on the left side of its containing block.
right	The element floats on the right side of its containing block.
none	Removes the float property from an element.

How Elements Float

A floated element is taken out of the normal flow and shifted to the left or right as far as possible in the space available of the containing element. Other elements normally flow around the floated items, unless they are prevented from doing so by their clear property. Elements are floated horizontally, which means that an element can only be floated left or right, not up or down.

```
img {  
    float: left;
```

```
}
```

If several floating elements are placed adjacently, they will float next to each other if there is horizontal room. If there is not enough room for the float, it is shifted downward until either it fits or there are no more floating elements present.

```
.thumbnail {  
    float: left;  
    width: 125px;  
    height: 125px;  
    margin: 10px;  
}
```

Module - 3

- 5. a. Discuss the advantage and disadvantages of client-side scripting. (08 Marks)**

Ans. Client-side scripts offer numerous advantages, including:

1. Allow for more interactivity by immediately responding to users' actions.
2. Execute quickly because they do not require a trip to the server.
3. May improve the usability of Web sites for users whose browsers support scripts.
4. Can give developers more control over the look and behaviour of their Web widgets.
5. Can be substituted with alternatives (for example, HTML) if users' browsers do not support scripts
6. Are reusable and obtainable from many free resources.

Client-side scripts also create additional worries which include:

1. Not all browsers support scripts, therefore, users might experience errors if no alternatives have been provided.
2. Different browsers and browser versions support scripts differently, thus more quality assurance testing is required.
3. More development time and effort might be required (if the scripts are not already available through other resources).
4. Developers have more control over the look and behaviour of their Web widgets; however, usability problems can arise if a Web widget looks like a standard control but behaves differently or vice-versa.

- b. Write a javascript code that displays text "VTU Belagavi" with increasing font size in the interval of 100ms in blue color, when the font size reaches 50pt it should stop. (08 Marks)**

Ans.

```

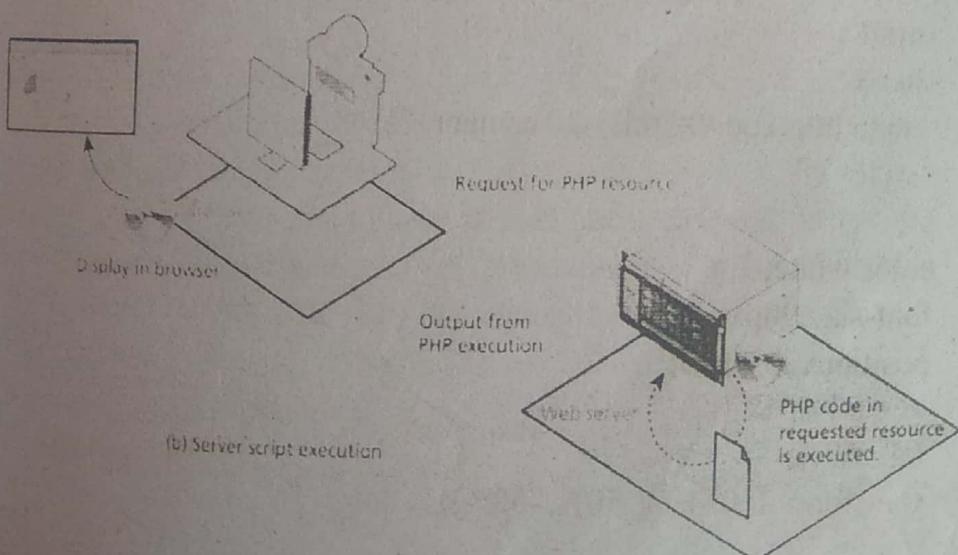
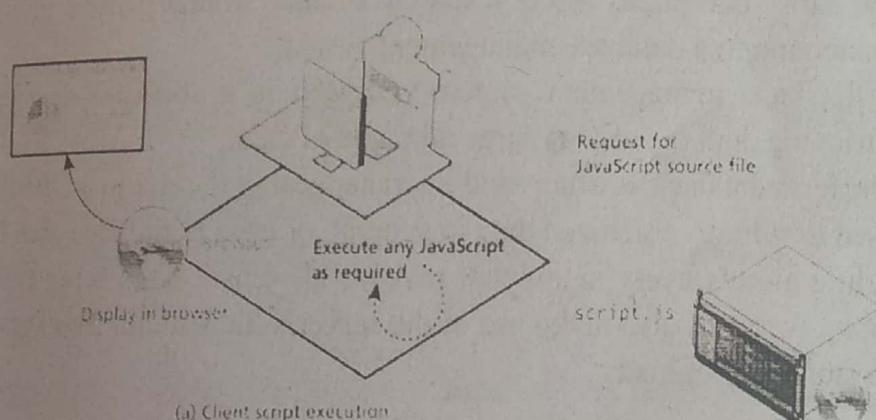
<html>
  <body>
    <p id="demo"></p>
    <script>
      var var1 = setInterval(inTimer, 1000);
      var size = 5;
      var ids = document.getElementById("demo");
      function inTimer() {
        ids.innerHTML = 'VTU Belagavi';
        ids.setAttribute('style', "font-size: " + size + "px; color: blue");
        size += 5;
        if(size >= 50 ) {
          clearInterval(var1);
        }
      }
    </script>
  </body>
</html>

```

OR

6. a. with a neat diagram, explain client and server script execution. (08 Marks)

Ans.



Client-Side

- The fundamental difference between client and server scripts is that in a client-side script the code is executed on the client browser.
- whereas in a server-side script, it is executed on the web server. clientside JavaScript code is downloaded to the client and is executed there.
- The server sends the JavaScript (that the user could look at), but you have no guarantee that the script will even execute.
- In contrast, server-side source code remains hidden from the client as it is processed on the server. The clients never get to see the code, just the HTML output from the script.
- Above Figure illustrates how client and server scripts differ. The location of the script also impacts what resources it can access. Server scripts cannot manipulate the HTML or DOM of a page in the client browser as is possible with client scripts.
- Conversely, a server script can access resources on the web server whereas the client cannot. Understanding where the scripts reside and what they can access is essential to writing quality web applications.

Server-Side

A server-side script can access any resources made available to it by the server. These resources can be categorized as data storage resources, web services, and software applications, as can be seen in Figure.

- The most commonly used resource is data storage, often in the form of a connection to a database management system.
- A database management system (DBMS) is a software system for storing, retrieving, and organizing large amounts of data.
- The term database is often used interchangeably to refer to a DBMS, but it is also used to refer to organized data in general, or even to the files used by the DBMS.
- While almost every significant real-world website uses some type of database, many websites also make use of the server's file system; for example, as a place to store user uploads.

b. Write a PHP program to greet the user based on time.

Ans. <html>

(08 Marks)

```

<head>
<meta http-equiv="refresh" content="1"/>
<style>
p {
color:white;
font-size:90px;
position: absolute;
top: 50%;
left: 50%;
transform: translate(-50%, -50%);
```

```
    }
    body{background-color:black;}
  </style>
<p>
<h1>Welcome User</h1>
<?php echo date(" h: i : s A");?>
</p>
</head>
</html>
```

Module - 4

1. a. Explain `$_GET` and `$_POST` superglobal arrays (08 Marks)

Ans. `$_GET`:

- `$_GET` is a super global variable used to collect data from the HTML form after submitting it.
- When form uses method get to transfer data, the data is visible in the query string, therefore the values are not hidden.
- `$_GET` super global array variable stores the values that come in the URL.
- Below is the HTML and PHP code to explain how `$_GET` works:

```
<!DOCTYPE html>
<html>
<head>
<title></title>
</head>
<body bgcolor="cyan">
<?php
  $name = $_GET['name'];
  $city = $_GET['city'];
  echo "<h1>This is ".$name." of ".$city."</h1><br>";
?
<img src = "2.jpg" alt = "nanilake" height = "400" width="500" />
</body>
</html>
```

`$_POST`:

- `$_POST` : It is a super global variable used to collect data from the HTML form after submitting it.
- When form uses method post to transfer data, the data is not visible in the query string, because of which security levels are maintained in this method.
- Below is the HTML and PHP code to explain how `$_POST` works:

```
<!DOCTYPE html>
<html>
<body>
<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
```

```

<label for="name">Please enter your name: </label>
<input name="name" type="text"><br>
<label for="age">Please enter your age: </label>
<input name="age" type="text"><br>
<input type="submit" value="Submit">
<button type="submit">SUBMIT</button>
</form>
<?php
$nm=$_POST['name'];
$age=$_POST['age'];
echo "<strong>".$nm." is $age years old.</strong>";
?>
</body>
</html>

```

- b. How do you read or write a file on the server from PHP? Give example

(08 Marks)

Ans. Read File on the server:

The fopen() function is also used to create a file. Maybe a little confusing, but in PHP, a file is created using the same function used to open files.

If you use fopen() on a file that does not exist, it will create it, given that the file is opened for writing (w) or appending (a).

The example below creates a new file called "testfile.txt". The file will be created in the same directory where the PHP code resides:

```

$myFile = " testfile.txt";
$fh = fopen($myFile, 'r');
$myFileContents = fread($fh, 21);
fclose($fh);
echo $myFileContents;

```

Writing File on the server:

The fwrite() function is used to write to a file.

The first parameter of fwrite() contains the name of the file to write to and the second parameter is the string to be written.

The example below writes a couple of names into a new file called "newfile.txt":

```

$myFile2 = "testFolder/ newfile.txt";
$myFileLink2 = fopen($myFile2, 'w+') or die("Can't open file.");
$newContents = "You wrote on me...";
fwrite($myFileLink2, $newContents);
fclose($myFileLink2);

```

OR

8. a. Write a PHP program to create a class STUDENT with the following specification.
Data Members: Name, RollNumber, AverageMarks
Member Functions: Read(getters) and write(setters)

```
Ans. <?php
class Books {
    /* Member variables */
    var $Name;
    var $RollNumber;
    var $Average;

    /* Member functions */
    function setName($par) {
        $this->Name = $par;
    }

    function getName() {
        echo $this->Name . "<br/>";
    }

    function setRollNumber($par) {
        $this->RollNumber = $par;
    }

    function getRollNumber() {
        echo $this->RollNumber . "<br/>";
    }

    function setAverageMarks($par) {
        $this->AverageMarks = $par;
    }

    function getAverageMarks() {
        echo $this->AverageMarks . "<br/>";
    }
}

?>
```

After creating your objects, you will be able to call member functions related to that object. One member function will be able to process member variable of related object only.

```
$Arjun->setName("Arjun");
$Arjun->setRollNumber("1XY15");
$Arjun->setAverageMarks("65.00");
```

```
$Arjun123->setName("Arjun");
```

```
$Arjun123->setRollNumber("I XY 15");
$Arjun123->setAverageMarks("65.00");

$Arjun->getName("Arjun");
$Arjun->getRollNumber("I XY 15");
$Arjun->getAverageMarks("65.00");

$Arjun123->getName("Arjun");
$Arjun123->getRollNumber("I XY 15");
$Arjun123->getAverageMarks("65.00");
```

b. How do you achieve data encapsulation in PHP? Give Example (08)

Ans. Encapsulation is just wrapping some data in an object. The term "encapsulation" is often used interchangeably with "information hiding"

```
<?php
```

```
class App {
    private static $_user;

    public function User( ) {
        if( $this->_user == null ) {
            $this->_user = new User();
        }
        return $this->_user;
    }
}

class User {
    private $_name;

    public function __construct() {
        $this->_name = "Joseph Crawford Jr.";
    }

    public function GetName() {
        return $this->_name;
    }
}

$app = new App();
echo $app->User()->GetName();
?>
```

9. a. What are HTTP cookies? How do you handle them in PHP? (08 marks)

Ans. A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.

Create Cookies With PHP

A cookie is created with the setcookie() function.

Syntax

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

Only the name parameter is required. All other parameters are optional.

PHP provides many ways to access cookies. Simplest way is to use either `$_COOKIE` or `$HTTP_COOKIE_VARS` variables.

Setting Cookies

Example:

```
<?php
    setcookie("name", "John Watkin", time() + 3600, "/", "", 0);
    setcookie("age", "36", time() + 3600, "/", "", 0);
?>
<html>
```

```
<head>
    <title>Setting Cookies with PHP</title>
</head>
```

```
<body>
    <?php echo "Set Cookies"?>
</body>
```

```
</html>
Accessing Cookies in PHP
```

```
<html>
```

```
<head>
    <title>Accessing Cookies with PHP</title>
</head>
```

```
<body>
```

```
<?php
    echo $_COOKIE["name"]. "<br />";
```

```

/* is equivalent to */
echo $HTTP_COOKIE_VARS["name"]. "<br />";

echo $_COOKIE["age"] . "<br />";

/* is equivalent to */
echo $HTTP_COOKIE_VARS["age"] . "<br />";
?>

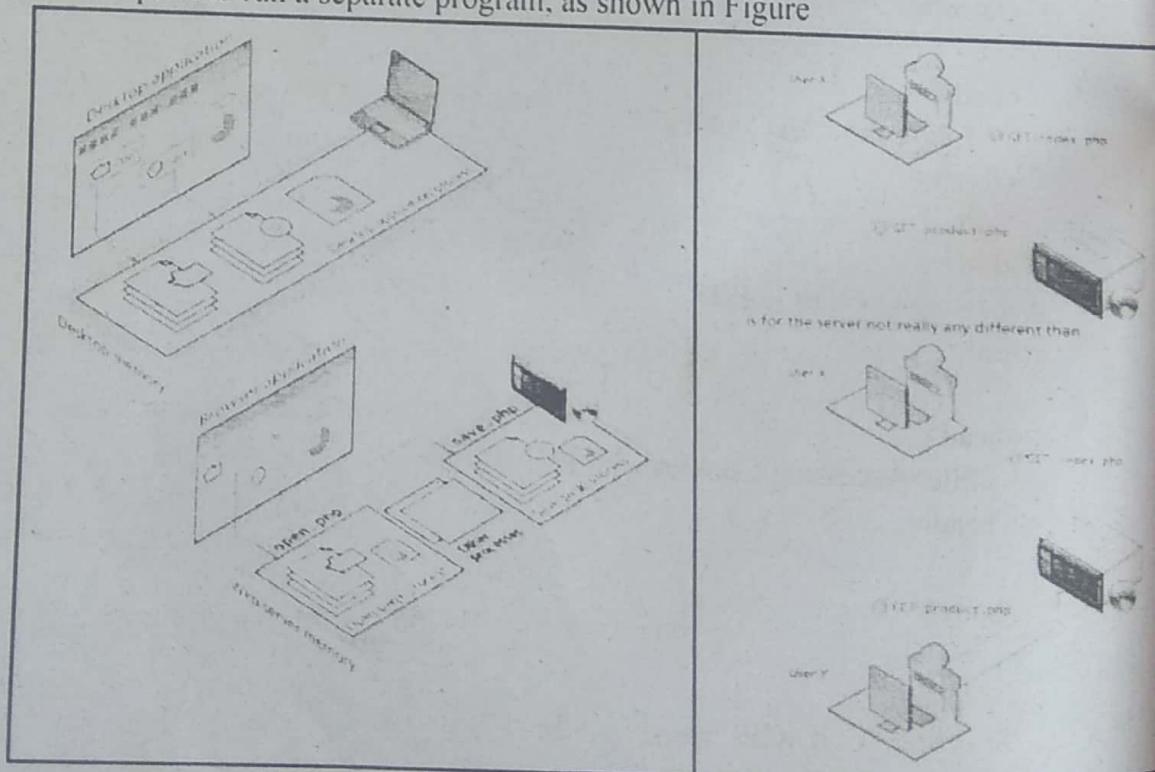
</body>
</html>

```

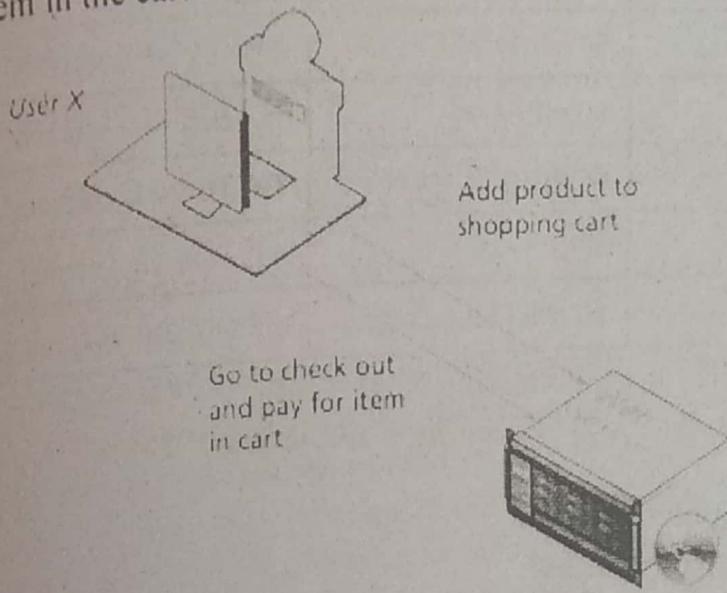
- b. Why is state is a problem for web applications? Explain. (08 Marks)**

Ans.

- Almost all applications need to process user inputs, output information, and read and write from databases or other storage media
- At first glance this problem does not seem especially formidable. Single-user desktop applications do not have this challenge at all because the program information for the user is stored in memory (or in external storage) and can thus be easily accessed throughout the application.
- Yet one must always remember that web applications differ from desktop applications in a fundamental way. Unlike the unified single process that is the typical desktop application, a web application consists of a series of disconnected HTTP requests to a web server where each request for a server page is essentially a request to run a separate program, as shown in Figure



- Furthermore, the web server sees only requests. The HTTP protocol does not, without programming intervention, distinguish two requests by one source from two requests from two different sources, as shown in below Figure
- While the HTTP protocol disconnects the user's identity from his or her requests, there are many occasions when we want the web server to connect requests together.
- Consider the scenario of a web shopping cart, as shown in Figure. In such a case, the user (and the website owner) most certainly wants the server to recognize that the request to add an item to the cart and the subsequent request to check out and pay for the item in the cart are connected to the same individual.



OR

10. a. What does \$() short and stand for in JQuery? Explain any 3 JQuery form selectors. (08 Marks)

Ans. The \$() factory function

JQuery selectors start with the dollar sign and parentheses – \$(). The factory function \$() makes use of following three building blocks while selecting elements in a given document –

Sr.No.	Selector & Description
1	Tag Name Represents a tag name available in the DOM. For example \$('p') selects all paragraphs <p> in the document.
2	Tag ID Represents a tag available with the given ID in the DOM. For example \$('#some-id') selects the single element in the document that has an ID of some-id.
3	Tag Class Represents a tag available with the given class in the DOM. For example \$('.some-class') selects all elements in the document that have a class of some-class.

Selector	Example	Selects
*	<code>\$('*')</code>	All elements
#id	<code>\$('#lastname')</code>	The element with id="lastname"
.class	<code>\$('.intro')</code>	All elements with class="intro"
.class,.class	<code>\$('.intro,.demo')</code>	All elements with the class "intro" or "demo"
element	<code>\$(“p”)</code>	All <p> elements
el1,el2,el3	<code>\$(“h1,div,p”)</code>	All <h1>, <div> and <p> elements
:first	<code>\$(“p:first”)</code>	The first <p> element
:last	<code>\$(“p:last”)</code>	The last <p> element
:even	<code>\$(“tr:even”)</code>	All even <tr> elements
:odd	<code>\$(“tr:odd”)</code>	All odd <tr> elements

b. Write DTD for the following XML Code:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<art>
  <painting id="290">
    <title>Balcony</title>
    <artist>
      <name>Manet</name>
      <nationality>France</nationality>
    </artist>
    <year>1868</year>
    <medium>Oil on canvas</medium>
  </painting>
</art>
```

(08 Marks)

Ans. <?xml version="1.0" encoding="ISO-8859-1"?>

```
<!DOCTYPE art [
  <!ELEMENT art (painting*)>
  <!ELEMENT painting (title,artist,year,medium)>
  <!ATTLIST painting id CDATA #REQUIRED>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT artist (name,nationality)>
  <!ELEMENT name (#PCDATA)>
  <!ELEMENT nationality (#PCDATA)>
  <!ELEMENT year (#PCDATA)>
  <!ELEMENT medium (#PCDATA)>
]>
<art>
  ...
</art>
```

Module - 1

1. a. What are the three main aims of HTML5 and expand the following
HTML, XML, DOCTYPE, PHP, WHATWG

(08 Marks)

Ans. The Goals of HTML5

HTML5 was born out of visible needs in the browser ecosystem, and the aims of its specifications are all responses to these needs. This section details the three most prominent goals of HTML5, which can be thought of as themes that you see throughout the book.

Improving the Native Web

According to the World Wide Web Consortium (W3C) specification, HTML5 "introduces markup and APIs for emerging idioms, such as Web applications." More specifically, HTML5 adds syntactic features to the Web that could previously only be accomplished with plug-ins. For instance, if serving video on the Web is a nearly ubiquitous expectation, web browsers ought to be able to accomplish it without additional help. The same goes for audio and other animated or dynamic content. Thus the `<audio>`, `<video>`, and `<canvas>` elements are some of HTML5's most important additions to the Web.

HTML5 doesn't just make plug-ins less necessary, it also increases the browser's functionality to be more in line with native mobile applications. Browser vendors and standards committees have begun work on application programming interfaces (APIs) that expose functionality of (mobile) devices within the browser. The most prominent example of this is the Geolocation API, which allows browsers to retrieve geographical location much like native phone apps do. There are several smaller niche APIs (such as one for device orientation) that also promise to afford more utility in the browser.

More Done with Less Code

One much more subtle feature of HTML5 is the ability to do more with less code. There are a lot of de facto standard web page features, such as placeholder text in forms, autofocusing on a particular input element once the page loads, client-side validation of form input, date and time pickers, and so on.

All of these concepts are considered standard-issue stuff on a modern web page, but every one of them requires at least a little bit of JavaScript to work. Because of this, these concepts are implemented across websites in many different ways, and are at times buggy or inconsistent with each other.

HTML5 simplifies these common design patterns (and more) by creating standardized ways to accomplish them in HTML alone. This empowers designers and also reduces

code maintenance and interoperability between platforms because the given feature's functionality can be more contextually handled by the browser.

The Semantic Web

The semantic web is a long-held dream of the Web's inventor, Tim Berners-Lee. He envisioned a web where content was not only readable by humans but also understood by machines. Just as we have to write carefully for humans to comprehend, it would also take a little footwork to make sure programs parsing web pages could pick up on meaningful content.

HTML – Hyper Text Markup Language

XML – eXtenstible Markup language

DOCTYPE – DOCumentType

PHP -Personal Home Page

WHATWG - Web Hypertext Application Technology Working Group

- b. List out the three types of lists and explain them with an example. (08 Marks)

Ans. There are three list types in HTML:

unordered list — used to group a set of related items in no particular order

ordered list — used to group a set of related items in a specific order

description list — used to display name/value pairs such as terms and definitions

Each list type has a specific purpose and meaning in a web page.

Unordered lists

Unordered (bulleted) lists are used when a set of items can be placed in any order. An example is a shopping list:

milk

bread

butter

coffee beans

Although the items are all part of one list, you could put the items in any order and the list would still make sense:

bread

coffee beans

milk

butter

Unordered list markup

Unordered lists use one set of `` tags wrapped around one or more sets of `` tags:

```
<ul>
  <li>bread</li>
  <li>coffee beans</li>
  <li>milk</li>
  <li>butter</li>
</ul>
```

Ordered lists

- Ordered (numbered) lists are used to display a list of items that should be in a specific order. An example would be cooking instructions:
1. Gather ingredients
 2. Mix ingredients together
 3. Place ingredients in a baking dish
 4. Bake in oven for an hour
 5. Remove from oven
 6. Allow to stand for ten minutes
 7. Serve

Ordered lists can be displayed with several sequencing options. The default in most browsers is decimal numbers, but there are others available:

Letters

Lowercase ascii letters (a, b, c...)
Uppercase ascii letters (A, B, C...).
Lowercase classical Greek: (é, ñ, í...)

Numbers

Decimal numbers (1, 2, 3...)
Decimal numbers with leading zeros (01, 02, 03...)
Lowercase Roman numerals (i, ii, iii...)
Uppercase Roman numerals (I, II, III...)
Traditional Georgian numbering (an, ban, gan...)
Traditional Armenian numbering (mek, yerku, yerek...)

As with unordered lists, you can use CSS to change the style of your ordered lists.
See Styling lists and links for more information.

Ordered list markup

Ordered lists use one set of `` tags wrapped around one or more sets of `` tags:

```
<ol>
  <li>Gather ingredients</li>
  <li>Mix ingredients together</li>
  <li>Place ingredients in a baking dish</li>
  <li>Bake in oven for an hour</li>
  <li>Remove from oven</li>
  <li>Allow to stand for ten minutes</li>
  <li>Serve</li>
</ol>
```

Description list markup

Description lists use one set of `<dl></dl>` tags wrapped around one or more groups of `<dt></dt>` (name) and `<dd></dd>` (value) tags. You must pair at least one `<dt></dt>` with at least one `<dd></dd>`, and the `<dt></dt>` should always come first in the source order. A simple description list of single names with single values would look like this:

```
<dl>
  <dt>Name</dt>
  <dd>Value</dd>
  <dt>Name</dt>
  <dd>Value</dd>
  <dt>Name</dt>
  <dd>Value</dd>
</dl>
```

This is rendered as follows:

Name

 Value

Name

 Value

Name

 Value

In the following example, we associate more than one value with a name, and vice versa:

```
<dl>
  <dt>Name1</dt>
  <dd>Value that applies to Name1</dd>
  <dt>Name2</dt>
  <dt>Name3</dt>
  <dd>Value that applies to both Name2 and Name3</dd>
  <dt>Name4</dt>
  <dd>One value that applies to Name4</dd>
  <dd>Another value that applies to Name4</dd>
</dl>
```

That code would render like this:

Name1

 Value that applies to Name1

Name2

Name3

 Value that applies to both Name2 and Name3

Name4

 One value that applies to Name4

 Another value that applies to Name4

OR

2. a. Define CSS and list out its benefits with the explanation.

(08 Marks)

Ans. CSS stands for Cascading Style Sheets

CSS describes how HTML elements are to be displayed on screen, paper, or in other media

CSS saves a lot of work. It can control the layout of multiple web pages all at once

External stylesheets are stored in CSS files

- **Improved control over formatting.** The degree of formatting control in CSS is significantly better than that provided in HTML. CSS gives web authors fine-grained control over the appearance of their web content.
- **Improved site maintainability.** Websites become significantly more maintainable because all formatting can be centralized into one CSS file, or a small handful of them. This allows you to make site-wide visual modifications by changing a single file.
- **Improved accessibility.** CSS-driven sites are more accessible. By keeping presentation out of the HTML, screen readers and other accessibility tools work better, thereby providing a significantly enriched experience for those reliant on accessibility tools.
- **Improved page download speed.** A site built using a centralized set of CSS files for all presentation will also be quicker to download because each individual HTML file will contain less style information and markup, and thus be smaller.
- **Improved output flexibility.** CSS can be used to adapt a page for different output media. This approach to CSS page design is often referred to as responsive design

b. What are selectors? List and explain selectors with an example. (08 Marks)

Ans. Refer Q. no 2b. of Dec-2018/Jan 2019

Module - 2

3. a. Apply the following table elements to display the following table:

Table elements: table, td, tr, th, tbody, tfoot, thead (08 Marks)

Ans. <!DOCTYPE html>

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

```
<h2>HTML Table</h2>
```

```
<table>
<thead>
<tr>
<th>SLNO</th>
<th>USN</th>
<th>Name</th>
<th>Dept</th>
</tr>
</thead>
<tbody>
<tr>
```

```

<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
<tfoot>
<tr>
<td>Total no. of rows</td>
<td>2</td>
<td>-</td>
<td>-</td>
</tr>
</tfoot>

</table>

</body>
</html>

```

- b. Explain with a neat diagram how form works? Discuss about query string and micro formats (08 Marks)**

Ans.

```

<form method="get" action="process.php">
  <fieldset>
    <legend>Details</legend>
    <p>
      <label>Title: </label>
      <input type="text" name="title" />
    </p>
    <p>
      <label>Country: </label>
      <select name="where">
        <option>Choose a country</option>
        <option>Canada</option>
        <option>Finland</option>
        <option>United States</option>
      </select>
    </p>
    <input type="submit" />
  </fieldset>
</form>

```

While forms are constructed with HTML elements, a form also requires some type of server-side resource that processes the user's form input as shown in Figure. The process begins with a request for an HTML page that contains some type of form on it. This could be something as complex as a user registration form or as simple as a search box. After the user fills out the form, there needs to be some mechanism for submitting the form data back to the server. This is typically achieved via a submit button, but through JavaScript, it is possible to submit form data using some other type of mechanism.

Because interaction between the browser and the web server is governed by the HTTP protocol, the form data must be sent to the server via a standard HTTP request. This request is typically some type of server side program that will process the form data in some way; this could include checking it for validity, storing it in a database, or sending it in an email.

Query String: The browser packages the user's data input into something called a query string. A query string is a series of name=value pairs separated by ampersands (the & character).

Microformats (sometimes abbreviated μF) are conventions used to embed semantics in HTML and provide an API to be used by search engines, aggregators, and other tools. These minimal patterns of HTML are used for marking entities that range from fundamental to domain-specific information, such as people, organizations, events, and locations.

Microformats are supported by all major search engines. The data is carried in the class property that can be added to any HTML element. In addition to being machine-readable, their format is designed to be easily read by humans.

OR

4. a. Identify the approaches to CSS layouts and explain them in detail (08 Marks)

Ans. Website Layout

A website is often divided into headers, menus, content and a footer:

Header

A header is usually located at the top of the website (or right below a top navigation menu). It often contains a logo or the website name.

```
.header {  
background-color: #F1F1F1;  
text-align: center;  
padding: 20px;  
}
```

Navigation Bar

A navigation bar contains a list of links to help visitors navigating through your website:

```
/* The navbar container */  
.topnav {  
overflow: hidden;
```

```
background-color: #333;
```

```
}
```

```
/* Navbar links */
```

```
.topnav a {  
    float: left;  
    display: block;  
    color: #f2f2f2;  
    text-align: center;  
    padding: 14px 16px;  
    text-decoration: none;
```

```
}
```

```
/* Links - change color on hover */
```

```
.topnav a:hover {  
    background-color: #ddd;  
    color: black;  
}
```

Content

The layout in this section, often depends on the target users. The most common layout is one (or combining them) of the following:

- 1-column (often used for mobile browsers)
- 2-column (often used for tablets and laptops)
- 3-column layout (only used for desktops)

```
/* Create three equal columns that floats next to each other */
```

```
.column {  
    float: left;  
    width: 33.33%;  
}
```

```
/* Clear floats after the columns */
```

```
.row:after {  
    content: " ";  
    display: table;  
    clear: both;  
}
```

```
/* Responsive layout - makes the three columns stack on top of each other  
instead of next to each other on smaller screens (600px wide or less) */
```

```
@media screen and (max-width: 600px) {  
    .column {  
        width: 100%;  
    }
```

Unequal Columns

The main content is the biggest and the most important part of your site. It is common with unequal column widths, so that most of the space is reserved for the main content. The side content (if any) is often used as an alternative navigation or to specify information relevant to the main content. Change the widths as you like, only remember that it should add up to 100% in total:

```
.column {  
    float: left;  
}  
  
/* Left and right column */  
.column.side {  
    width: 25%;  
}  
  
/* Middle column */  
.column.middle {  
    width: 50%;  
}  
  
/* Responsive layout - makes the three columns stack on top of each other  
instead of next to each other */  
@media screen and (max-width: 600px) {  
.column.side, .column.middle {  
    width: 100%;  
}  
}  
}  
  
Footer
```

The footer is placed at the bottom of your page. It often contains information like copyright and contact info:

```
.footer {  
    background-color: #F1F1F1;  
    text-align: center;  
    padding: 10px;  
}
```

Responsive Website Layout

By using some of the CSS code above, we have created a responsive website layout, which varies between two columns and full-width columns depending on screen width:

b. What is responsive design? Why its important? Explain in detail (08 Marks)

Ans. Essentially, responsive design is a way to put together a website so that it automatically scales its content and elements to match the screen size on which it is viewed. It keeps images from being larger than the screen width, and prevents visitors on mobile devices from needing to do extra work to read your content.

The ultimate goal of responsive design is to avoid the unnecessary resizing, scrolling, zooming, or panning that occurs with sites that have not been optimized for different devices. It is often very difficult to navigate these sites, and it may even cost you potential customers who become frustrated with trying to figure out how to do something.

The benefits of choosing responsive design for your website include:

1. Cost effectiveness
2. Flexibility
3. Improved user experience
4. Search engine optimization gains
5. Ease of management

1. COST EFFECTIVENESS

Maintaining separate sites for your mobile and non-mobile audiences can get expensive. By using responsive design, you can save money by eliminating the cost of paying for a mobile site. You will only need to invest in a single site design to appeal to all visitors and all devices.

2. FLEXIBILITY

When you have a website with responsive design, you can make changes quickly and easily. You do not need to worry about making changes on two websites. This flexibility is a huge advantage when you just want to make a quick design tweak or fix a typo on your site—you only have to do it once.

3. IMPROVED USER EXPERIENCE

User experience is crucial to website owners. You want people to like your site, and you want it to be easy to use to convince them to come back. If someone visits your website on a mobile device, and it takes forever to load or your pictures do not have the proper resolution, it can make your company appear unprofessional.

4. SEARCH ENGINE OPTIMIZATION GAINS

Search engine optimization, or SEO, is a strategy used by many companies to help boost themselves in Google's search page rankings. The closer you are to the top, the better the chance potential customers will find you.

Responsive design can help with SEO because Google, as mentioned, gives preference to websites that are mobile-friendly. In combination with other SEO factors, responsiveness can help give you a big boost in search engine results.

5. EASE OF MANAGEMENT

Most businesses, especially smaller ones, don't have a lot of time to update or refresh the way their website looks. But rather than having to hire a designer to handle every aspect of your website, responsive design allows you to make the changes yourself, quickly and easily.

5 a. What is javascript and listener? Discuss the advantages and disadvantages of client side scripting. (08 Marks)

Ans. Javascript:

- JavaScript is the Programming Language for the Web.
- JavaScript can update and change both HTML and CSS
- JavaScript can calculate, manipulate and validate data

Listener:

An event listener is a procedure or function in a computer program that waits for an event to occur. Examples of an event are the user clicking or moving the mouse, pressing a key on the keyboard, disk I/O, network activity, or an internal timer or interrupt. The listener is programmed to react to an input or signal by calling the event's handler.

The term event listener is often specific to Java and JavaScript. In other languages, a subroutine that performs a similar function is referred to as an event handler.

Advantages

Client-side scripts offer numerous advantages, including:

- Allow for more interactivity by immediately responding to users' actions.
- Execute quickly because they do not require a trip to the server.
- May improve the usability of Web sites for users whose browsers support scripts.
- Can give developers more control over the look and behaviour of their Web widgets.
- Can be substituted with alternatives (for example, HTML) if users' browsers do not support scripts
- Are reusable and obtainable from many free resources.

Disadvantages

Client-side scripts also create additional worries which include:

- Not all browsers support scripts, therefore, users might experience errors if no alternatives have been provided.
- Different browsers and browser versions support scripts differently, thus more quality assurance testing is required.
- More development time and effort might be required (if the scripts are not already available through other resources).
- Developers have more control over the look and behaviour of their Web widgets; however, usability problems can arise if a Web widget looks like a standard control but behaves differently or vice-versa.

b. What are s/w layers? What are benefits do they provide? Explain in detail

(08 Marks)

Ans. When designing software to solve a problem, it is often helpful to abstract the solution a little bit to help build a cognitive model in your mind that you can then implement. Perhaps the most common way of articulating such a cognitive model is via the term layer. In object-oriented programming, a software layer is a way

of conceptually grouping programming classes that have similar functionality and dependencies. Common software design layer names include:

- Presentation layer. Classes focused on the user interface.
- Business layer. Classes that model real-world entities, such as customers, products, and sales.
- Data layer. Classes that handle the interaction with the data sources.

Presentation Layer

This type of programming focuses on the display of information. JavaScript can alter the HTML of a page, which results in a change, visible to the user. These presentation layer applications include common things like creating, hiding, and showing divs, using tabs to show multiple views, or having arrows to page through result sets. This layer is most closely related to the user experience and the most visible to the end user.

Validation Layer

JavaScript can be also used to validate logical aspects of the user's experience. This could include, for example, validating a form to make sure the email entered is valid before sending it along. It is often used in conjunction with the presentation layer to create a coherent user experience, where a message to the presentation layer highlights bad fields. Both layers exist on the client machine, although the intention is to prevalidate forms before making transmissions back to the server.

OR

6. a. Compare the server-side technologies in detail

(08 Marks)

- Ans.
1. Insertion of continuously changing content into a web page, for example weather or stock quotes. Also, any arbitrary logic can be used to determine certain content will be shown or not. This purpose and (10) below are the primary purposes of server-side scripting.
 2. Authentication, authorization and session tracking - although rudimentary authentication and authorization is supported by most Web servers, anything more than the "BASIC" http authentication and ACLs (access control lists) over static resources requires server-side programs. Similarly, handling cookies and keeping information about the session and/or the user is best handled by server-side scripting.
 3. Template-driven page generation. Including repeated content like header/footers and navigation menus around the "content area" of a web page.
 4. Personalization and customization of content based on authentication and authorization defined above in (2). This also includes the serving of content based on the content of the page (e.g. ads) or the browsing behavior of the user.
 5. Dynamic image generation, e.g. page counters, human-readable characters for security, maps, overlays etc.
 6. Dynamic generation of CSS and Javascript.
 7. Generating and reading HTTP headers. Although web servers provide rudimentary abilities, server-side scripting can best generate cache control and other complex headers.

8. Handling POST form input - accepting the input of a form and writing it to storage (file system, database, session etc.). This also includes business transaction commitment control (ALL or NONE) and input error handling.
9. Device mapping - generating different types of content (HTML, XML, WML) based on the user agent that sent the HTTP request.
10. Retrieval of data in response to query string parameters and insertion into a web page. This is perhaps the most common purpose of utilizing scripting in generating content as part of a GET request. e.g. sports statistics, staff list, downloadable files list etc. The data can be retrieved from a database, file system or other forms of storage.
11. Communication with other programs, libraries and APIs - e.g. sending out e-mail, handling message queues, LDAP etc.
12. Re-use of persistent business objects. HTTP is stateless, but the setup and tear-down of business objects has a very high overhead in terms of time and server resources. Server-side scripting allows us to interact with such re-usable business objects e.g. application servers, EJBs, .NET services and Web services.

b. Write a php program to demonstrate the session. Program: Store page view count on refresh

Ans. Sessions:

```
<?php
session_start();
?
<html>
<body>
<?php
$_SESSION["user"] = "Sachin";
echo "Session information are set successfully.<br/>";
?>
<a href="session2.php">Visit next page</a>
</body>
</html>
```

Page Count on Refresh :

```
<?php
session_start();
if(isset($_SESSION['views']))
$_SESSION['views'] = $_SESSION['views']+1;
else
$_SESSION['views']=1;
echo"views = ".$_SESSION['views'];
?>
```

Module - 4

- Ques.** What are super global arrays? List and explain.

(08 Marks)

- Ans.** PHP Global Variables - Superglobals

Several predefined variables in PHP are "superglobals", which means that they are always accessible, regardless of scope - and you can access them from any function, class or file without having to do anything special.

The PHP superglobal variables are:

1. \$GLOBALS
2. \$_SERVER
3. \$_REQUEST
4. \$_POST
5. \$_GET
6. \$_FILES
7. \$_ENV
8. \$_COOKIE
9. \$_SESSION

PHP \$GLOBALS

\$GLOBALS is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods). PHP stores all global variables in an array called \$GLOBALS[index]. The index holds the name of the variable.

PHP \$_SERVER

\$_SERVER is a PHP super global variable which holds information about headers, paths, and script locations.

PHP \$_REQUEST

PHP \$_REQUEST is used to collect data after submitting an HTML form.

PHP \$_POST

PHP \$_POST is widely used to collect form data after submitting an HTML form with method="post". \$_POST is also widely used to pass variables.

PHP \$_GET

PHP \$_GET can also be used to collect form data after submitting an HTML form with method="get".

\$_GET can also collect data sent in the URL.

- Ques.** Define constructor and discuss the concepts of inheritance, polymorphism and object interface with respect to OOP.

(08 Marks)

- Ans.** PHP allows developers to declare constructor methods for classes. Classes which have a constructor method call this method on each newly-created object, so it is suitable for any initialization that the object may need before it is used.

```
<pre>
class BaseClass {
    public __construct() {
        echo "Base class constructor";
    }
}</pre>
```

```
}

class SubClass extends BaseClass {
    function __construct() {
        parent::__construct();
        print "In SubClass constructor\n";
    }
}

class OtherSubClass extends BaseClass {
    // inherits BaseClass's constructor
}

// In BaseClass constructor
$obj = new BaseClass();

// In BaseClass constructor
// In SubClass constructor
$obj = new SubClass();

// In BaseClass constructor
$obj = new OtherSubClass();
```

> Inheritance is a well-established programming principle, and PHP makes use of this principle in its object model. This principle will affect the way many classes and objects relate to one another.

For example, when you extend a class, the subclass inherits all of the public and protected methods from the parent class. Unless a class overrides those methods, they will retain their original functionality.

This is useful for defining and abstracting functionality, and permits the implementation of additional functionality in similar objects without the need to reimplement all of the shared functionality.

Polymorphism is derived from two Greek words. Poly (meaning many) and morph (meaning forms). Polymorphism is one of the PHP Object Oriented Programming (OOP) features. In general, polymorphism means the ability to have many forms. If we say it in other words, "Polymorphism describes a pattern in Object Oriented Programming in which a class has varying functionality while sharing a common interfaces."

There are two types of Polymorphism; they are:

1. Compile time (function overloading)
2. Run time (function overriding)

Object interfaces allow you to create code which specifies which methods a class must implement, without having to define how these methods are implemented. Interfaces are defined in the same way as a class, but with the interface keyword replacing the class keyword and without any of the methods having their contents defined.

All methods declared in an interface must be public; this is the nature of an interface. Note that it is possible to declare a constructor in an interface, what can be useful in some contexts, e.g. for use by factories.

OR

8. a. Explain with an example, the two basic technique for read/write files in PHP. (08 Marks)

Ans. Refer 7 b. of Dec-2018/Jan-2019

- b. Write the PHP code to validate phone number (08 Marks)

Ans. \$phone = "+91-444-444-5555";
if(validate_phone_number(\$phone) == true) {
echo "Phone number is valid";
} else {
echo "Invalid phone number";
}

```
function validate_phone_number($phone)
{
    // Allow +, - and . in phone number
    $filtered_phone_number = filter_var($phone, FILTER_SANITIZE_NUMBER_INT);
    // Remove "-" from number
    $phone_to_check = str_replace("-", "", $filtered_phone_number);
    // Check the lenght of number
    // This can be customized if you want phone number from a specific country
    if(strlen($phone_to_check) < 10 || strlen($phone_to_check) > 14) {
        return false;
    } else {
        return true;
    }
}
```

Module - 5

9. a. What are cookie? What Is the purpose of it? Demonstrate cookie with PHP program. (08 Marks)

Ans. Refer Q. no 9 a of Dec-2018/Jan-2019

- b. Discuss jQuery selectors in detail (08 Marks)

Ans. Refer 10 a. of Dec-2018/Jan-2019

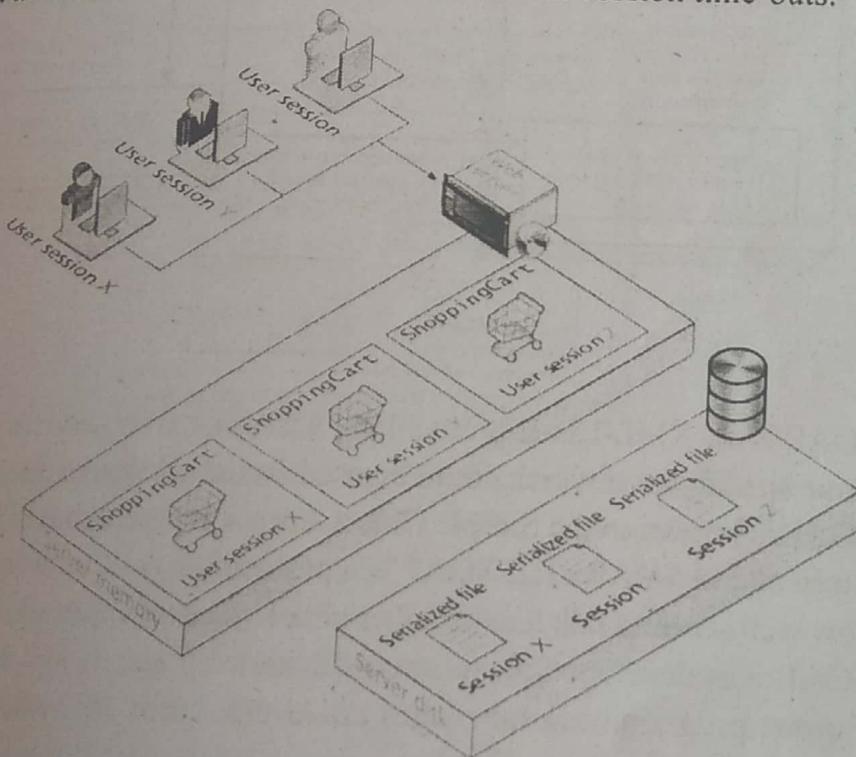
10. a. Discuss the following:
 i) Session Cookies
 ii) Persistent cookies
 iii) Session state

(08 Marks)

Ans. i) **Session cookies** - these are temporary cookie files, which are erased when you close your browser. When you restart your browser and go back to the site that created the cookie, the website will not recognize you. You will have to log back in (if login is required) or select your preferences/themes again if the site uses these features. A new session cookie will be generated, which will store your browsing information and will be active until you leave the site and close your browser. More on session cookies.

ii) **Persistent cookies** – these files stay in one of your browser's subfolders until you delete them manually or your browser deletes them based on the duration period contained within the persistent cookie's file (more on persistent cookies).

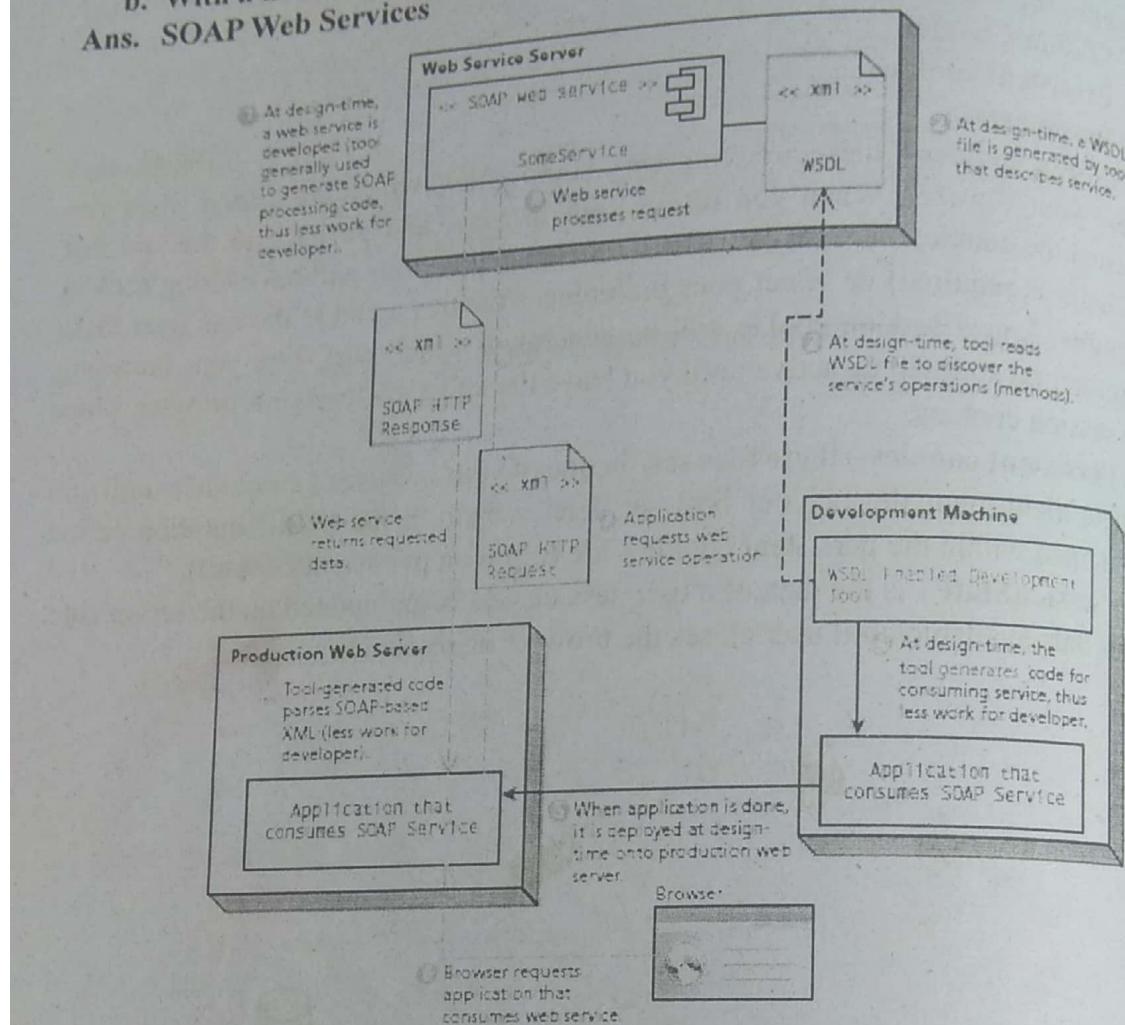
iii) **SessionState** - is the data of a user session and is maintained on the server side. This data available until user closes the browser or session time-outs.



Session state is ideal for storing more complex (but not too complex . . . more on that later) objects or data structures that are associated with a user session. The classic example is a shopping cart. While shopping carts could be implemented via cookies or query string parameters, it would be quite complex and cumbersome to do so. In PHP, session state is available to the developer as a superglobal associative array, much like the `$_GET`, `$_POST`, and `$_COOKIE` arrays.³ It can be accessed via the `$_SESSION` variable, but unlike the other superglobals, you have to take additional steps in your own code in order to use the `$_SESSION` superglobal.

b. With a neat diagram explain SOAP and REST web services

Ans. SOAP Web Services



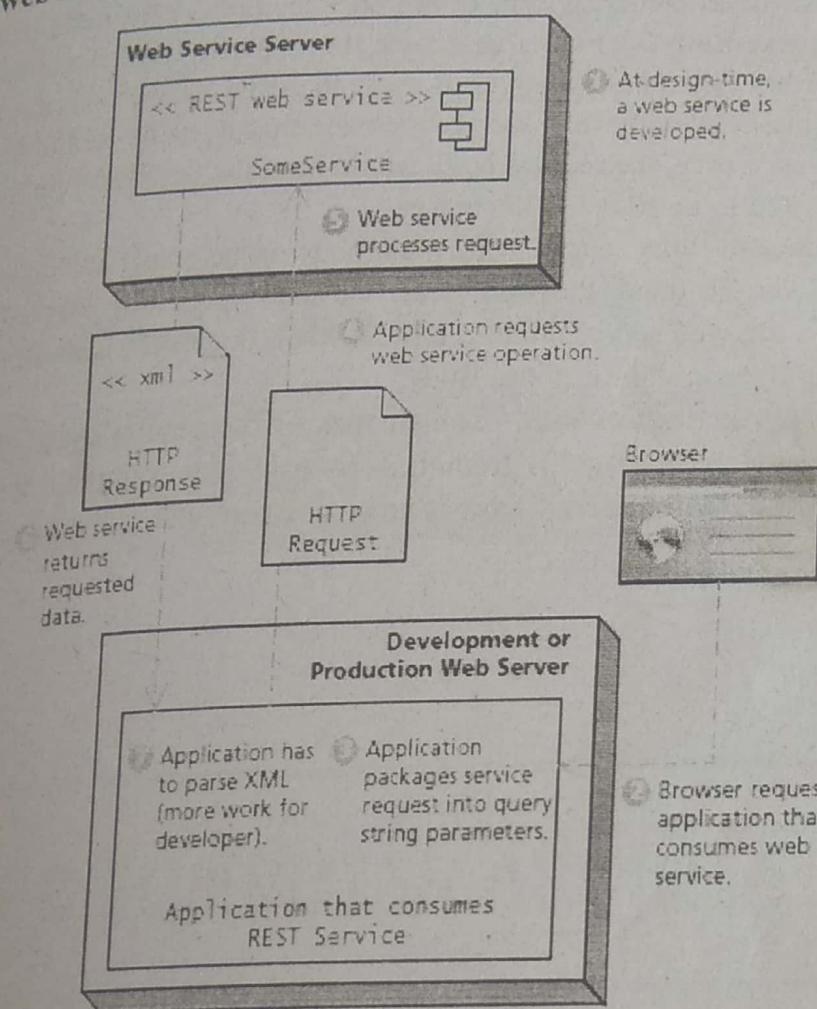
- SOAP is an XML-based protocol for accessing web services over HTTP. It has some specification which could be used across all applications.
- SOAP is known as the Simple Object Access Protocol, but in later times was just shortened to SOAP v1.2. SOAP is a protocol or in other words is a definition of how web services talk to each other or talk to client applications that invoke them.
- SOAP was developed as an intermediate language so that applications built on various programming languages could talk easily to each other and avoid the extreme development effort.

The SOAP message is nothing but a mere XML document which has the below components.

- An Envelope element that identifies the XML document as a SOAP message – This is the containing part of the SOAP message and is used to encapsulate all the details in the SOAP message. This is the root element in the SOAP message.
- A Header element that contains header information – The header element can contain information such as authentication credentials which can be used by the calling application. It can also contain the definition of complex types which could be used in the SOAP message. By default, the SOAP message can contain

parameters which could be of simple types such as strings and numbers, but can also be a complex object type.

REST web service



REST is used to build Web services that are lightweight, maintainable, and scalable in nature. A service which is built on the REST architecture is called a RESTful service. The underlying protocol for REST is HTTP, which is the basic web protocol. REST stands for REpresentational State Transfer

The key elements of a RESTful implementation are as follows:

- Resources** – The first key element is the resource itself. Let assume that a web application on a server has records of several employees. Let's assume the URL of the web application is <http://www.google.com>. Now in order to access an employee record resource via REST, one can issue the command <http://www.google.com/gmail/> - This command tells the web server to please provide the details of the gmail.
- Request Verbs** - These describe what you want to do with the resource. A browser issues a GET verb to instruct the endpoint it wants to get data. However, there are many other verbs available including things like POST, PUT, and DELETE. So in the case of the example <http://www.google.com/l>, the web browser is

actually issuing a GET Verb because it wants to get the details of the employee record.

3. **Request Headers** – These are additional instructions sent with the request. These might define the type of response required or the authorization details.
4. **Request Body** - Data is sent with the request. Data is normally sent in the request when a POST request is made to the REST web service. In a POST call, the client actually tells the web service that it wants to add a resource to the server. Hence, the request body would have the details of the resource which is required to be added to the server.
5. **Response Body** – This is the main body of the response. So in our example, if we were to query the web server via the request <http://www.google.com/about/def>, the web server might return an XML document with all the details of the employee in the Response Body.
6. **Response Status codes** – These codes are the general codes which are returned along with the response from the web server. An example is the code 200 which is normally returned if there is no error when returning a response to the client.