WEB TECHNOLOGY & ITS APPLICATIONS.

Sem : VI

SubCode : 18CS63

MODEL QUESTION PAPER.

MODULE – 1.

1.a) What are 3 aims of HTML5? [4M]

b) Explain the need of cascade in CSS? Explain the 3 principles of cascade with suitable CSS script segments. [8M]

c) Explain 2 types of URL Referencing techniques with suitable scripts in HTML 5. [8M]

OR.

2. a) List & Explain different selectors available in CSS [8M]

b) Discuss the HTML5 semantic structure elements. [8M]

c) List the different text properties with a description. [4M]

MODULE – 2.

3. a) Explain different Form Widgets created with the <input> tag. [8M]

b) Write HTML code for following table.

DAY	SEMINAR SCHEDULE		SEMINAR TOPIC
	BEGIN	END	
MONDAY	8:00 am	5:00 pm	INTRODUCTION TO XML Validity & DTD & NS.
TUESDAY	11:00 am	2:00 pm	XPAT 4
	11:00 am	2:00 pm	XSL Transformations
	2:00 pm	5:00 pm	
WEDNESDAY	8:00 am	5:00 pm	XSL Formatting Objects

OR

4. a) Explain liquid layout design for websites with an example. List the liquid layout benefits & limitations. [8M]

b) Explain different ways of positioning elements in css layout techniques. [8M]

c) What are the importance of responsive design? Explain Briefly. [4M]

MODULE-3

5. a) Write JS code that displays text "CORONA VIRUS" with increasing font-size in the interval of 100ms in blue color, when font size reaches 50pt in teal color & should stop. [8M]

b) Explain the advantages & disadvantages of client side scripting. [6M]

c) With suitable diagram, explain APACHE modules in PHP. [6M]

OR

6. a) With suitable code segment, explain 2 approaches for event handling in JS. [8M]

b) Write PHP program to greet the user based on time. [8M]

c) Explain 2 methods in JS to access DOM nodes with examples. [4M]

MODULE-4

7. a) List & Explain different superglobal arrays. [8M]

b) Explain the different error handling methods, with suitable code segments. [8M]

c) How do you read or write file on server from PHP? Give Examples. [4M]

OR

8.a) Write PHP Program to create a class Employee with the following specifications: [8M]

Data members : Name, ID, payment.

Member functions : Read (getters) & write (Setters).

use the above specifications to read & print the information of 10 students.

b) Explain the support for inheritance in PHP with UML class diagram. [6M]

c) Explain 3 approaches to restrict file size in file upload with suitable code segments. [6M]

MODULE - 5

9.a) Explain different types of caching need to improve performance of web applications. [8M]

b) With suitable PHP script, Explain loading & procesing an XML document in JavaScript. [8M]

c) Explain creating & reading cookies with suitable PHP scripts. [4M]

OR

10.a) Define AJAX. Explain AJAX request by writing UML Diagram. [8M]

b) Explain JS pseudo-classes with examples. [8M]

c) Explain converting a JSON string to JSON object in Javascript with suitable code segment. [4M]

* * * * *

PROF. FARZANA. N. NADAF.

SEM : VI SUB : WEB TECHNOLOGY & ITS APPLICATION

SUBCODE : 18CS63.

1.a) There are 3 main aims to HTML5 :-

i) Specify unambiguously how browsers should deal with invalid markup.

ii) Provide an open, nonproprietary programming framework for creating rich web applications.

iii) Be backwards compatible with existing web.

b) Cascade refers in CSS to conflicting rule; precedence of child elements.

* There are 3 principles of Cascading are :-

i) Inheritance :- Many CSS properties affect not only themselves but their descendants also.

ii) Specificity :- Which style rule takes precedance when more than one style rule could be applied to the same element.

iii) Location :- The principle of location is that when rules have the same specificity, then the latest are given more weight.

Eg :-

```
<head>                                                    (1)
    <link rel="stylesheet" href="stylesA.css" />          (2)
    <link rel="stylesheet" href="styleww.css" />
    <style>                                               (4)
        #example { color : orange; }                      (5) overrides
                 color : magenta;
    </style>                                              (6) overrides
</head> <body>
    <p id="#emple" style="color : red;">
```

c) URL Referencing technique are :-

i) Relative Referencing :- We need to successfully reference file within website. It requires a Relative referencing.

ii) ABSOLUTE REFERENCING :- When referencing a page or resource on an external site, a full ABSOLUTE REFERENCING is required. i.e

a) protocol (http://) b) domain name

c) any path d) filename of desired resource.

Eg:- <a href = "example.html">
<a href = "css/images/back.gif">
<a href = "../images/about.html">
<a href = "../../css/images/back.gif">

2.a) The different selectors available in Css are :-

* Element selectors
* Id Selectors
* Psendo Element & " class selectors
* Class Selectors
* Attribute "
* Contextual Selectors.

* Element Selectors ⇒
```
p { margin : 0;
    padding : 0; }
```

* Class Selectors ⇒
```
.first { font-style : italic;
         color : red; }
```

* ID Selectors ⇒
```
#Comm { margin : 0;
        padding : 0; }
```

* ATTRIBUTE Selectors ⇒
```
[title] { border-bottom : 2px dotted;
          text-decoration : underline;
        }
```

* Psendo Selectors ⇒
```
a:link { color : blue; }
a:active { background-color : teal;
```

* Contextual Selectors :- #main div p:first-child
{ color : green ; }

b) HTML5 Semantic Structure Elements are :-

* HEADER & FOOTER :- It contains the headings for a section, along with content such a material. Footer contains less important material such as navigation, copyright notices.

* HEADING GROUPS :- <hgroup> element can be used in contexts other than header.

<HEADER> <HGROUP> <H1> --- </H1> </HGROUP>

* NAVIGATION :- <nav> element represents a section of page that contains links to other pages.

* ARTICLES & SECTIONS :- Articles element represents a section of content that forms an independent part of a document. Section element represents section of a document.

* FIGURE & FIGURE CAPTION :- <FIGURE> element repⁿ some flow content, optionally with a caption. <FIGCAPTION> used to annotate illustration, diagrams, photos, code listings.

* ASIDE :- It is similar to <figure> element i.e used for making up content i.e separate from main content on page.

Eg :- <BODY>

<HEADER>
<HGROUP> ...
</HGROUP>
<nav> --- </nav>
</HEADER>

<SECTION> --- </SECTION>
<article> --- </article>
<figure>

<figcaption> --- </figcaption>
</figure>
<aside> --- </aside>
<footer> --- </footer>

c) The different text-properties are :-

a) text-decoration.
b) text-align
e) line-height
c) letter-spacing
d) text-decoration.
f) word-spacing.

3) a) Form Widgets created are <inpnt> are :-

text : creates single line text entry box.
<inpnt type="text" name="a" />.

textarea : creates multiple line text entry box.
<textarea rows="3" ---7.

password : single-line text with bullets/some characters.
<inpnt type="password" />.

search : single-line text entry suitable for search string. <inpnt type="search" ---- />.

email : which takes an email address.
<inpnt type="email" ---- />.

tel : which takes an entering a telephone number.
<inpnt type="tel" ---- />

url : creates a single line entering URL.
<inpnt type="url" ---- />.

b) <HTML>
<BODY> <TABLE BORDER="1" CAPTION="Diff LANG">
<TR> <TH ROWSPAN="3">. DAY </TH>
<TH COLSPAN="3">. SEMINAR </TH> </TR>
<TR> <TD COLSPAN="2"> SCHEDULE </TD>
<TD ROWSPAN="2"> TOPICS </TD> </TR>.
<TR> <TD> BEGIN </TD> <TD> END </TD> </TR>.
<TR> <TD> MONDAY </TD> <TD> 8:00 AM </TD>
<TD> 5:00 pm </TD> <TD ROWSPAN="2">. INTRODUCTION
TO XML </TD> </TR>.
<TR> <TD> VALIDATION : DTD & NS </TD> </TR>

```
<TR> <TD ROWSPAN="3"> TUESDAY </TD>
    <TD> 11:00 AM </TD><TD> 2:00 pm </TD>
    <TD ROWSPAN="2"> XPATH </TD> </TR>
<TR> <TD> 11:00 am </TD><TD> 2:00 pm </TD></TR>
<TR> <TD> 2:00 PM</TD> <TD> 5:00 pm </TD>
    <TD> XSL transformations </TD> </TR>
<TR> <TD> WEDNESDAY </TD> <TD> 8:00 am </TD>
    <TD> 5:00 pm </TD> <TD> XSL Formating
        Objects </TD> </TR> . </TABLE>
</BODY> </HTML>
```

4.a) The LIQUID LAYOUT :-

* It's deal with the problem of multiple screen sizes. It is also called FLUID LAYOUT.
* It adapts to different browser sizes so there is neither wasted white space nor any need for Horizontal scrolling.
* DisAdvantages :- i) Difficult to create coz some elements, such as images, have fixed pixel sizes. ii) As screen grows or shrinks dramatically, in that line length may become too long or too short.
iii) Creating a usable liquid layout is generally more difficult than creating a fixed layout.

ADVANTAGES :- BENEFITS :-

* Liquid Layout is that it adapts to different browser sizes, So there is neither wasted white space nor any need for horizontal scrolling.
* The % values in CSS are a % of current browser width, so layout in which all widths are expressed as % should adapt to any browser size.

b) The different ways of positioning elements in CSS layouts techniques. are :-

* Relative Positioning :-
It is an element displaced out of its normal flow position & moved relative to where it would have been placed.

* ABSOLUTE POSITIONING :-
It is removed completely from the normal flow. The space is not left for the moved element, as it is no longer in the normal flow.

eg :-
```
figcaption { background-color : #EDEDDD;
             padding : 5px; width : 150px;
             position : absolute;
           }
figure { top : 150px; left : 200px;
         position : relative; }
```

* Z-INDEX :- Each positioned element has a stacking order defined by the z-index property. Items closest to viewer have a larger z-index value,

Eg:- figure {
position: absolute;
top: 150px;
left: 400px;
z-index: 1;
}

figcaption {
position: absolute;
top: 90px;
left: 140 px;
z-index: -1;
}

* Fixed Position :-

It is a type of absolute positioning, except the positioning values are in relation to viewport.

* It is commonly used to ensure that navigation elements or advertisements are always visible.

eg:- figure { position: fixed; top: 0px; left: 0px; }

4.c) The importances of responsive design are :-

* Liquid layouts ①
* Setting viewports via <meta> tag ②
③ * Scaling images to viewport size ⇒ img {max-width: 100%; }
④ * Customizing CSS for different viewports using media queries.

② <meta name="viewport" content="width=device-width" />

④ eg:- @media only screen and (max-width: 480 px) { ----- }

① It is the one which most elements have their width specified in %.

img { max-width: 100 %; }

<BODY>

5. a) <script type="text/javascript">

```
var v1 = setInterval(inTimer, 1000);
var fs = 5;  var ids = document.getElementById("demo");
function inTimer()
{
  ids.innerHTML = "CORONA VIRUS";
  ids.setAttribute('style', "font-size:"+fs+"px;
                             color:blue");
  fs+=5;
  if(fs >= 50) { clearInterval(v1);
                 v2 = setInterval(deTimer, 1000);
  }
}
function deTimer()
{
  fs -= 5;
  ids.setAttribute('style', "font-size:"+fs+"px;
                             color:Teal");
  if(fs === 5) {
    clearInterval(v2);
  }
}
```

</script> </BODY> </HTML>

b) Client-Side Scripting :-

Advantages :-

* Processing can be offloaded from the server to client machines, ∴ reducing the load on server.
* Browser can respond more rapidly to user events than a request to remote server, which improves user experience.
* JS can interact with downloaded HTML in a way that server can't; Creating a user experience more like desktop s/w than simple HTML.

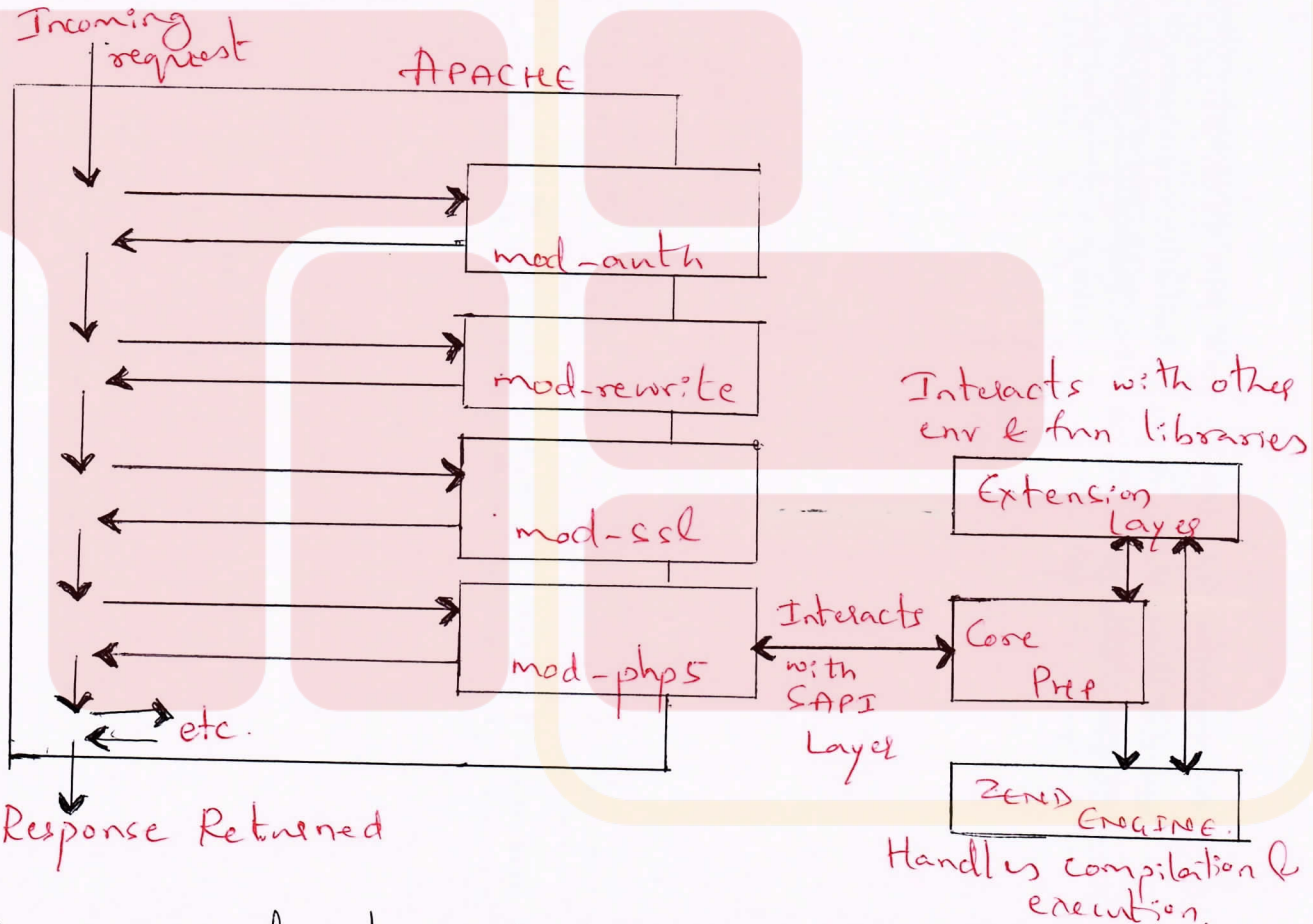
*Disadvantages of Client-side Scripting:-

⇒ There is no guarantee that the client has JS enabled, it means required functionality must be hosted on server.

⇒ The idiosyncrasies between various browsers & OS make it difficult to test for client configuration.

⇒ Javascript heavy web-applications can be complicated to debug & maintain.

c) APACHE Modules in PHP:-



* Diagram indicates that; PHP usually installed as an APACHE module.

* PHP_module mod_php5 is sometimes referred to as SAPI layer. (Server Application Prog. Interface).

* SAPI handles the interaction between PHP & web server environment.

*Zend Engine is virtual machine that processes & executes Prep files. It also handles memory mangement garbage collection & dispatching fun^n calls to modules outside of Prep.

6. a) The 2 approaches for event handling in JS are:-

*Inline - Event Handler :-

⇒ JS events allow programmer to react to user interactions.

⇒ inline JS calls are intuitive.

eg:-
click for pop-up

⇒ When a user clicks the , event is triggered & alert is executed.

*LISTENER APPROACH :-

There are 2 functions to add an event & delete an event.

⇒ Main advantage of this approach is that the code can be written anywhere, including an external file.

⇒ Limitation is only one handler can respond to any given element event.

eg:- var greet = document.getElementById('exp1');
greet.addEventListener('click', alert('GoodMor));
greet.addEventListener('mouseout', alert('GoodBye'));
— —
— —
greet.removeEventListener( );

b) <?php

```
$t = date('H');
$t2 = date('e');
if($t <="12") { echo "Good Morning"; }
else if ($t >="12" && $t <"17")
{ echo "Good Afternoon"; }
else echo "Good Night"; }
?>
```

c) The two methods in JS to access DOM nodes are :-

* getElementId : using id of a element, we can access the element data. Since ID must be a unique in an HTML document. It returns a single node rather than a set of results.

eg:-
```
var v1 = document.getElementById("demo");
v1.style.display = "block";
v1.style.border = "solid 1px";
```

* getElementByTagname :- It returns a Nodelist of elements whose tagname matches the passed name parameter.

eg:-
```
var mylink = document.getElementsByTagName("p");
for(i=0; i<mylink.length; i++)
{ if(mylink[i].className == "std_class") {
    mylink[i].onClick = function(){
        this.stylebackgroundcolor = "#foo"; }
}
}
```

7. a) The superglobal arrays are :-

$GLOBALS, $_COOKIES, $_ENV, $_FILES, $_GET, $_POST, $REQUEST, $_SESSION, $_SERVER.

$GLOBALS :- array for storing data that needs superglobal scope.

$_COOKIES :- array of cookie data passed to page via HTTP request. It gives more information about user logged-in, how many times hit the pages.

$_ENV :- Server environment data.

$_FILES :- file items uploaded to server

$_GET :- Array of Query string data passed to server via the URL. The interaction with user input can be accessed by the methods GET & POST of HTML form elements.

$_POST :- Array of Query string data passed to server via HTTP header.

$_SERVER :- It contains a variety of information. It some of the info contained within HTTP request headers sent by the client.

eg:- `echo $_SERVER["SERVER_NAME"]."<br/>";`
`echo $_SERVER['SERVER_SOFTWARE"]."<br/>";`

b) The different error handling methods are :-

* PROCEDURAL ERROR HANDLING :-

⇒ Programmer needs to explicitly test for error conditions after performing a task might generate an error.

* It may result in a great deal of code duplication.

Eg:- `$conn = mysqli_connect(DBHOST, DBNAME, DBPASS, DBNAME);`
`$error = mysqli_connect_error();`
`if($error != null) { ---- }`

* OBJECT-ORIENTED EXCEPTION HANDLING :-

⇒ When a runtime error occurs, PHP throws an exception. This exception can be caught & handled either by the function, class or page that generated the exception or by code that called function or class.

⇒ PHP uses the try-catch progⁿ construct to progⁿtically deal with exception at runtime.

Eg:-
```
function throwException($msg=null, $code=null)
  { throw new Exception($msg, $code); }
try {
    $conn = mysqli_connect( ---- ), or
    throwException("error");
}
catch(Exception e) {
    echo 'Caught exception on line '.
        $e->getLine();
}
finally { --- }
```

c) There are 2 technique for read/write file in PHP.

* Stream Access :- to read a small portion of the file at a time. It is the most memory efficient approach when reading large files.

* To read a content from file, we use fgets(), fopen(), fread() & fgetsc().

eg:- while ($line = fgets ($f)) { $ln++; }
pf("%2d", $ln); }

All-In-Memory :-

⇒ We can read the entire file into memory.
⇒ It is easier to use, at cost of relinquishing fine-grained control.

eg:- $f = file_get_contents (FILENAME);
file_put_contents (FILENAME, $w);

e.g.
```
<?php
class EMPLOYEE {
        private $name;
        private $ID;
        public $pay;

    function __construct($fn, $id, $pay){
        $this->setName($fn);
        $this->setID($id);
        $this->setPay($pay);    }

    public function getName() { return $this->name; }
public function getID() { return $this->ID; }
public function getPay() { return $this->pay; }
public function setName($n) { $this->name = $n; }
public function setID($d) { $this->ID = $d; }
public function setPay($p) { $this->pay = $p; }
}

EMPLOYEE[] a = new EMPLOYEE[10];
    $a[0] = new EMPLOYEE("Foo", 2124, 70,000);
    $a[1] = new EMPLOYEE("Mohit", 9584, 80,000);
    - - -
echo "EMPLOYEE INFO \n";
for ($i=1; $i<=10; $i++) { echo $a[$i]->outputAsTable(); }
}. }
```

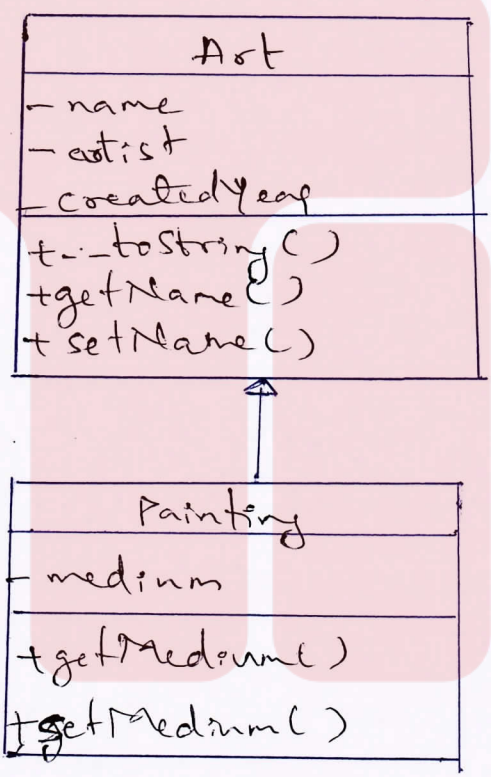
b) INHERITANCE :-

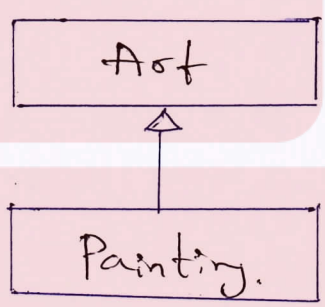
* It is the key concept of Object oriented design & programming. It allows us to create new PHP classes that reuse, extend & modify the behavior of a defined class in another PHP class.

* PHP allows only one class to inherit at a time.

* PHP class is defined as a subclass by using the extends keyword.

class Painting extends Art { .... }





eg:- class Art
{
— —
}

class Painting extends Art { public function foo() {
$p = new Painting();
$w = parent :: getName();
$x = parent :: getOriginal();
}

c) The 3 approaches to restrict file size in file upload are :-

* via HTML ① in input form
* via JS in input ② form.
* PHP coding ③.

Eg:- 1)
```
foreach($_FILES as $ffk => $ffArr)
{
    if ($ffArr["error"] != UPLOAD_ERR_OK)
    {
        echo "Error". $ffk. "has error" . $ffArr["error"];
    }
}
else { echo $ffk. "uploaded successfully"; }
```

* Limiting upload file size via HTML.

```
<HTML><FORM enctype="multipart/form-data' method='post'>
<input type="hidden" name="MAX_FILE_SIZE"
                                value="1000000" />
<input type='file' name='f1'/>
<input type='submit' /> </form>.
```

* Limiting upload file size via JS

```
<script>
var file = document.getElementById('file');
var max_size = document.getElementById("max_file_size").value;
if (file.files && file.files.length == 1) {
    if (file.files[0].size > max_size ) {
        alert ("file must b less than" +(max_size/1024) + "KB");
        e.preventDefault (); } } </script>.
```

* Limiting upload file size via PHP.

```
$max_file_size = 10000000;
foreach ( $_files as $ffk => $ffArr) {
    if($ffArr["size"] > $max_file_size) {
        echo "Error". $ffk. " is too big"; }
printf("%s %s %.2f KB" $ffArr["size"]/1024);
```

9.) The different types of caching used to improve performance of web applications are :-

* Page O/P Caching :- which saves the rendered output of a page or user control & reuses the output instead of reprocessing the page when a user requests the page again.

* It is useful for pages whose content does not change frequently but which require significant processing to create.

* 2 models are :- full page caching
partial page caching.

* The full page caching ; entire contents of a page are cached.

* In partial page caching, only specific parts of page are cached while other parts are dynamically generated in normal manner.

* APPLICATION DATA CACHING :-

⇒ Limitation of page output caching is performance gains only if entire page is same for numerous requests

⇒ It is a page, which will place commonly used collections of data require time-intensive queries from DB or web server into cache memory.

⇒ A widely available free PECL extension called memcache is used to provide this caching.

⇒ It should be stressed memcache should not be used to store large collections.

b) Loading & Procesing an XML document via JS.

eg :-
```
<script>
if(window.XMLHttpRequest){
    xmlHttp=new XMLHttpRequest();}
else{ xmlHttp=new ActiveXObject("Microsoft.XMLHTTP")
}
xmlHttp.open("GET","arto.xml",false);
xmlHttp.send(); xmlDoc=xmlHttp.responseXML;
paintings=xmlDoc.getElementsByTagName("painting");
if(paintings){ for(i=0;i<paintings.length;i++)
{ alert("id="+paintings[i].getAttribute("id"));
  title=paintings[i].getElementsByTagName("title");
  if(title){ alert("title="+title[0].textContent);
  }
}
</script>
```

* XML procesing in PHP is divided into 2 basic styles :-

⇒ IN-MEMORY APPROACH :- which involves reading the entire XML file into memory into some type of data structure with functions for accesing & manipulating the data.

⇒ EVENT or PULL APPROACH :- which reading a file in a few elements or lines at a time, therefore avoiding the memory load of large XML files.

⇒ JS loads the entire document into memory where it is transformed into a hierarchical tree data structure.

⇒ JS supports a variety of node traversal fun^n

as well as properties for accessing information within an XML node.

c) PHP provides mechanisms for writing & reading cookies. PHP cookies are created using the setcookie() function & retrieved using the $_COOKIES superglobal associative array.

eg:-
```
<?php
$etm = time() + 60*60*24;
$nm = "username"; $v = "Mohit";
setCookie($nm, $v, $etm); ?>
```

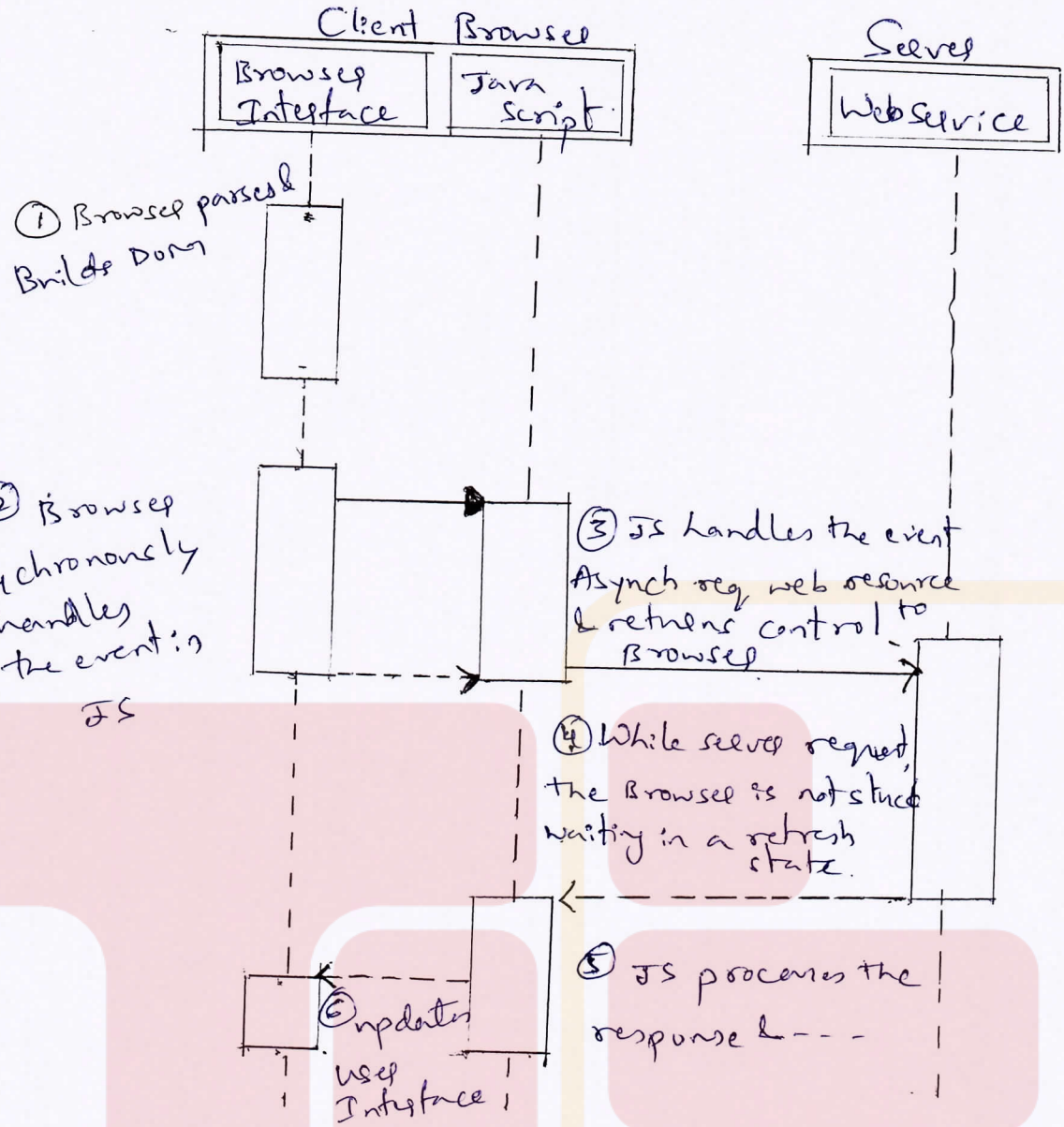
* Before reading a cookie, we must check to ensure that the cookie exists. In PHP, If cookie has expired, then client Browser would not send anything & $_COOKIE array would be blank.

```
<?php if(!isset($_COOKIE['username']))
{ // no valid Cookie }
else { echo "Username Retrieved from Cookie:";
echo $_COOKIE['username']; }
?>
```

10. a) Asynchronous JS with XML (AJAX) is term used describe a paradigm that allows a web browser to send messages back to server without interrupting the flow.

⇒ The UML sequence diagram of AJAX request

*Responses to asynchronously requests are caught in JS as events. These events subsequently trigger in UI or make additional requests.



* The difference from the typical sychronous requests which require the entire page to refresh in response to a request.

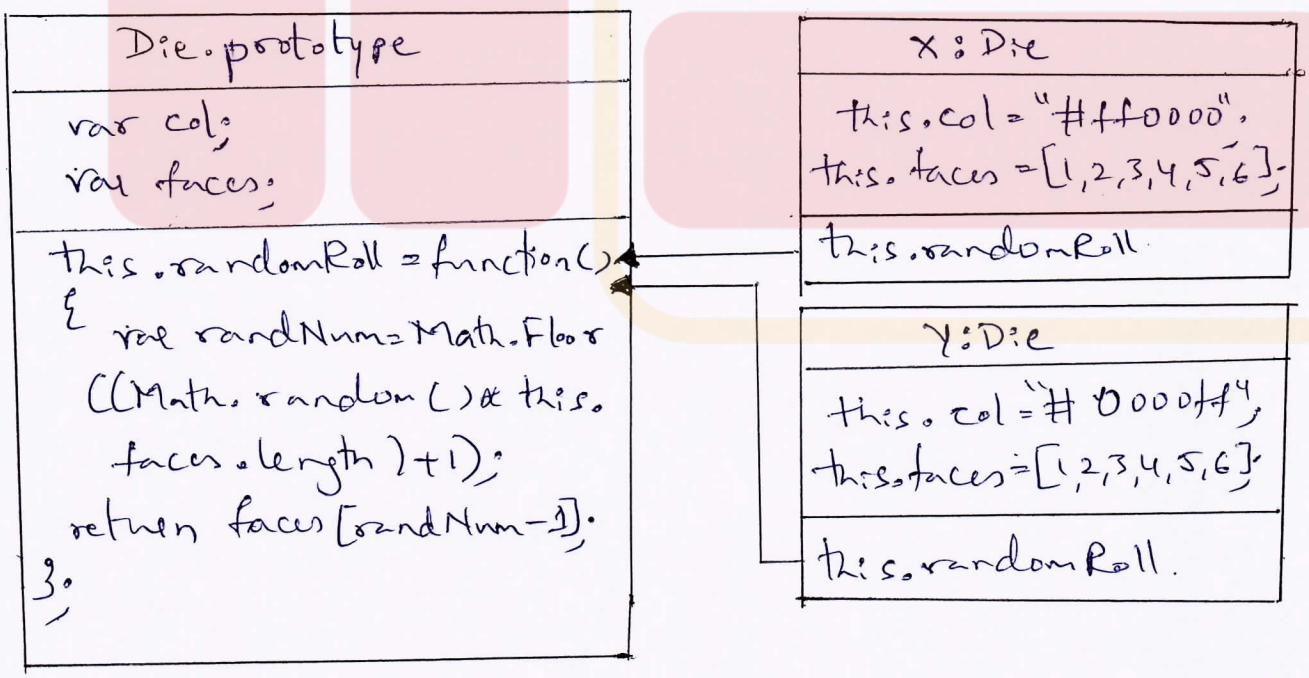
b) JS doesn't support most of the object oriented features. Instead it define psendo-classes. through a variety of interesting & nonintuitive syntax.

⇒ JS include increased code sense, better memory management & easier maintainance.

⇒ Example to difine the psendo-class with an internally defined method.

```
function Die(col) {
    this.color = col;
    this.faces = [1,2,3,4,5,6];
}
Die.prototype.
    .randomRoll = function() {
        var randNum = Math.floor((Math.random()*this.faces.
                      length +1);
        return faces[randNum-1];
    };
};
```

* Adding method inside of a class definition is by assigning an anonymous function to a variable.
* Prototype are an essential syntax mechanism in JS, & used to make JS behave more like OO Language.
* The prototype properties & methods are defined once for all instances of an object.
* Figure illustrates JS prototypes as pseudo-classes.



c) Converting a JSON string into a PHP object is straightforward:

→ Ex:-
```
<?php
    $text = '{"artist":{"name":"Foo","nationality":"INDIAN"}}';
    $anObj = json_decode($text);
    echo $anObj->artist->nationality;
    $anArr = json_decode($text, true);
    echo $anArr['artist']['nationality']; ?>
```

* JSON-decode function return either PHP Object or an associative array.
* JSON data is coming from external source, we should always check for parse errors via the json_last_error() function.

```
<?php
    $text = '{"artist":{"name":"Foo",
              "nationality":"INDIAN"}}';
    $anObj = json_decode($text);
    if(json_last_error() == JSON_ERROR_NONE)
    {
        echo $anObj->artist->nationality; }
?>
```

CBCS SCHEME

USN

Seventh Semester B.E. Degree Examination, July/August 2021
Web Technology and It's Applications

Time: 3 hrs. Max. Marks: 100

Note: Answer any FIVE full questions.

1 a. What are HTML Elements and Attributes? Explain. **(06 Marks)**
b. Explain the different Relative Link Type Referencing with suitable example. **(08 Marks)**
c. What is CSS? List and explain benefits of CSS. **(06 Marks)**

2 a. Explain ordered and unordered list with example. **(05 Marks)**
b. With an example explain CSS Box model. **(08 Marks)**
c. List the different selectors available in CSS and explain in detail. **(07 Marks)**

3 a. Create a table that correctly uses the caption, thead, tfoot and tbody elements. Briefly discuss the role of each of these elements. **(10 Marks)**
b. What is a responsive design? Why it is important? **(05 Marks)**
c. Explain how rowspan and colspan attributes are used? **(05 Marks)**

4 a. Describe how block level elements are different from inline elements. Be sure to describe any two different types of inline elements with simple example. **(10 Marks)**
b. In what situations would you use a radio button and a checkbox? With an example explain briefly. **(05 Marks)**
c. Explain the role of CSS preprocessors in the web development workflow. **(05 Marks)**

5 a. Define software layer? Explain the various common software design layers in Javascript with a neat diagram. **(08 Marks)**
b. What are form events in Javascript? List and explain different form events. **(05 Marks)**
c. Demonstrate the use of inline, external and embedded Javascript with an example for each. **(07 Marks)**

6 a. What is Fail-Safe design and why does it matters? **(04 Marks)**
b. Explain Document Object Model. Demonstrate the DOM tree with an example. **(08 Marks)**
c. What are server-side include files? Why are they important in PHP? **(08 Marks)**

7 a. What are the superglobal arrays in PHP? What function is used to determine if a value was sent via Query string? **(10 Marks)**
b. How do you read or write a file on the server from PHP? Explain with suitable example. **(10 Marks)**

Important Note : 1. On completing your answers, compulsorily draw diagonal cross lines on the remaining blank pages.
2. Any revealing of identification, appeal to evaluator and /or equations written eg, 42+8 = 50, will be treated as malpractice.

SET_C00

8 a. Define Class and Object. Interpret the concept of data encapsulation, Inheritance, Polymorphism and Object interface with respect to OOP. **(10 Marks)**

b. Explain _construct( ) and _destruct( ) with example or each. **(10 Marks)**

9 a. What are HTTP Cookies? How do you handle them in PHP? **(08 Marks)**

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

c. What does $() shorthand stand for in jQuery? **(04 Marks)**

10 a. Explain how sessions stored between requests. **(05 Marks)**

b. Write a jQuery selector to get all the <P> that contain the word "Hello". **(05 Marks)**

c. What are the commonly used animations in jQuery? Explain with suitable example. **(10 Marks)**

* * * * *

KLS's VDIT HALIYAL.

DEPT. OF CSE.

SUB: WEB TECHNOLOGY & ITS APPLICATIONS. [18CS63]

1.a) HTML Elements are called as Tags of a webpage. HTML documents are composed of textual content & HTML Elements. HTML element is more expensive term which includes a element name within angle brackets. [6M]

* HTML element is identified in document by tags. It consists of element name within angle brackets.

* HTML elements contain attributes. An attribute is a name:value pair that provides more information abt HTML element.

Ex:- VDIT

Element name — attribute — HyperText — closing Tag

<body> This is some text .

child. <h1> Title goes here </h1>.

sibling.   This is important

 .

Ancestor.

</body>

b) The different Relative link Type References are:- [8M]

* Same Directory :-

<a href="example.html">.

* child Directory :

<a href="images/logo.gif" />.

* Grandchild Directory
<a href="css/images/background.gif">
* Parent Directory
<a href="../about.html">
* Sibling Directory
<a href="../images/about.html">
* Root Reference
<a href="/images/background.gif">
* Default Document.
<a href="members">

c) Cascading Style Sheets is a W3C standard for describing the appearance of HTML elements. It can be added directly to HTML element within <head> element or a separate text file contain only CSS. [6M]

Benefits of CSS :-

* Improved Control over formatting :-
CSS gives web authors to fine grained control over the appearance of their web content.
* Improved Site maintainability :-
It allows us to make site-wide visual modifications by changing a single file
* Improved Accessibility :-
CSS driven sites are more accessible.
* Improved page download speed :
It allows to be quicker to download cos of size of file.
* Improved output flexibility :-
It can be used to adopt a page for different output media.

2.a) Ordered List :-

Collections of items have set of order, it has default values as in browser in a numbered list. [SM]

```
<OL>
  <LI> Introduction </LI>
  <LI> BackGround </LI>
  <LI> My Solution </LI>
  <LI> <OL> <LI> METHODOLOGY </LI>
            <LI> Results </LI>
            <LI> Discussion </LI>
       </OL> </LI>
  <LI> Conclusion </LI>
</OL>
```

o/p ⇒

1. Introduction
2. Background
3. My Solution
   1. Methodology
   2. Results
   3. Discussion
4. Conclusion

* Unordered List :-

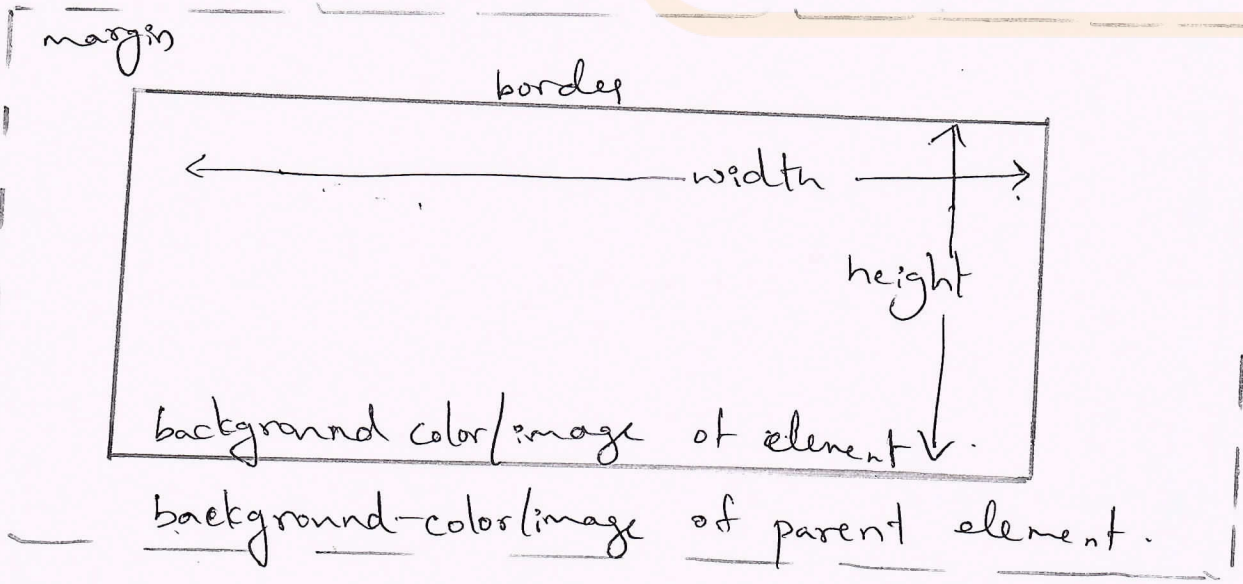
Collections of items in no particular order ∴ default rendered by browser is bulleted List.

```
<UL> <LI><a href="index.html">Home</a></LI>
  <LI> ABOUT US </LI>
  <LI> Products </LI>
  <LI> Contact us </LI>
</UL>
```

output ⇒

- Home
- About us
- Products
- Contact us.

b) All HTML elements exist within an Element Box. [SM]



*There are different background properties :-
-background, background-attachment, background-color, background-image, background-size, background-repeat.

*Borders :- It provide a way to visually separate elements. We can put 4 sides of an element.

eg:- body { background : white url(./images/logo.gif) norepeat
background-position : 300px 50px ;
} border-top-color : red ;

*Margins & Padding :- These are essential properties for adding white space to web page, which can differentiate one element from another.

eg:- p { border : solid 4pt teal ;
margin : 0px ; width : 200px ;
} padding : 10px ; height : 100px ;

c) The different selectors are :-

*Element Selectors :- selects all instances of given HTML elements.

eg:- p, div, aside { margin : 0 ;
padding : 0 ;
}

p { margin : 10px ;
} padding : 10px ;

*.class Selectors :-
Allows us to target different HTML elements regardless of their position in document tree.

eg:- <style>
.first { font-style : italic ;
</style> } color : red ;

— —
— —
<h1 class="first">—</h1>

*ID SELECTORS :-
Allows to target a specific element by its id attribute or its type or position.

* It can have the form with pound(#) followed by id name.

eg:- <style> #com{ color:red;} }
</style>.
<div id="com">.
<p> Easy on HDR buddy </p> </div>.

* Attribute Selectors:-

It provides a way to select HTML elements either by presence of an element attribute.

eg:- <style>
[title] { cursor:help;
padding-bottom: 3px;
border-bottom: 2px dotted blue;
text-decoration: none;
</style>}

<div>
- -- = - </div>.

* Pseudo-Element & Pseudo-class Selectors:-

It is a way to select explicitly as an element in HTML document tree. Pseudo-class selector to variety of family relationships.

eg:- a:link { text-decoration: underline;
color: blue; }
a:visited { text-decoration: none;
font-weight: bold; }.
a:active { background-color: yellow; }.

* Contextual Selectors:-

It allows to select elements based on their ancestors, descendants or siblings.

 Canada
 Germany   .
<div id="main"> — -- </div>.
<div> <p> By <time> Oct 1 2015 </time> </p> </div>.

3.a) <Caption> element is used to add the heading of the table. <thead> ⇒ Table header potentially include other <tr> elements. <tfoot> which can be added before the body. <tbody> ⇒ which display the actual content of a web page. [10M]

```
<table>
  <caption> Foench Paintings </caption>
  <thead> <tr> <th> Title </th>
               <th> Artist </th> <th> Year </th> </tr>
  </thead>
  <tfoot> <tr> <td colspan="2"> Total No. of painting </td>
               <td> 2 </td> </tr> </tfoot>
  <tbody> <tr> <td> Death of Marat </td>
               <td> Jacques - David </td>
               <td> 1793 </td> </tr>
  </tbody> </table>;
```

---- // create two or more rows.

b). The responsive design, which responds to changes in the browser size that go beyond the width [5M] scaling of a liquid layout.

* One of the main problems of a liquid layout is that images & horizontal navigation elements tend to be fixed size & when the browser window shrinks to size of a mobile browser, liquid layouts can become unusable.

* So In responsive design, images will be scaled down & navigation elements will be replaced as browser shrinks. ∴ Responsive Design is very important.

c) The rowspan & colspan can be used in a element attribute. There are 2 things about tables.

* First is all content must appear within <td> or <th> container. Second is each row must have same no. of <td> or <th> containers. [5M]

eg:- <table> <tr> <th> Title </th> <th> Artist </th>
<th> Year </th> <th colspan="2"> Size </th> </tr>
<tr> <td> Death of Marat </td> <td> David </td>
<td> 1793 </td> <td> 162cm </td> <td> 128cm </td>
<tr> - - - </table>.

4.a) Block-level elements such as , , <h2> & <table> are each contained on their own line. Because block-level elements begin with a line break, without styling, two block-level elements can't exist on same line. [10M]

* It use CSS box model with margin, paddings, background colors & borders.
* Inline-elements don't form their own blocks but are displayed within lines. Normal text in an HTML document is inline, as are elements such as , <a>, <img> & .
* There are two different types of inline elements:-
→ Replaced inline Elements. → Non replaced "

eg:- This photo of Pond in Central Park, New York city was taken on <time> Oct 22 2015. </time> .

* Replaced inline elements are elements whose content & appearance is defined by some external resources. such as <img> & various form elements.
* These have width & height defined by external resource.
* Non-replaced inline elements are the elements whose content is defined within the document, which includes all other inline elements.

b) Radio Buttons are useful when the user want to select a single item from a small list of choices & we want all the choices to be visible.

* Checkboxes are used for getting Yes/No or ON/OFF responses from the user. We can group checkboxes together by having them share the same name attribute.

Eg:- <input type="radio" name="who" value="1"> North America
<input type="radio" name="who" value="2"> South America
<input type="radio" name="who" value="3" checked> Asia
.
<label> I accept the s/w licence </label>
<input type="checkbox" name="visit" value="Canada">Canada
<input type="checkbox" name="visit" value="france">France
<input type="checkbox" name="visit" value="Germ">Germany
.

c) CSS preprocessors :-
These are the tools which allow developers to write CSS which takes advantages of programming ideas such as variables, inheritance, calculations &

& functions.

* A CSS preprocessor which takes code written in some type of preprocessed language & converts that code into normal CSS.
* One of the best way to use preprocessor of a CSS is with colors.
* CSS preprocessors such as LESS, SASS & stylus provide these type of functionality.

eg:-

```
$colorSchemeA : #796d6d;
footer { background-color : $colorSchemeA;
}
```

↓

SASS Preprocessor

↓

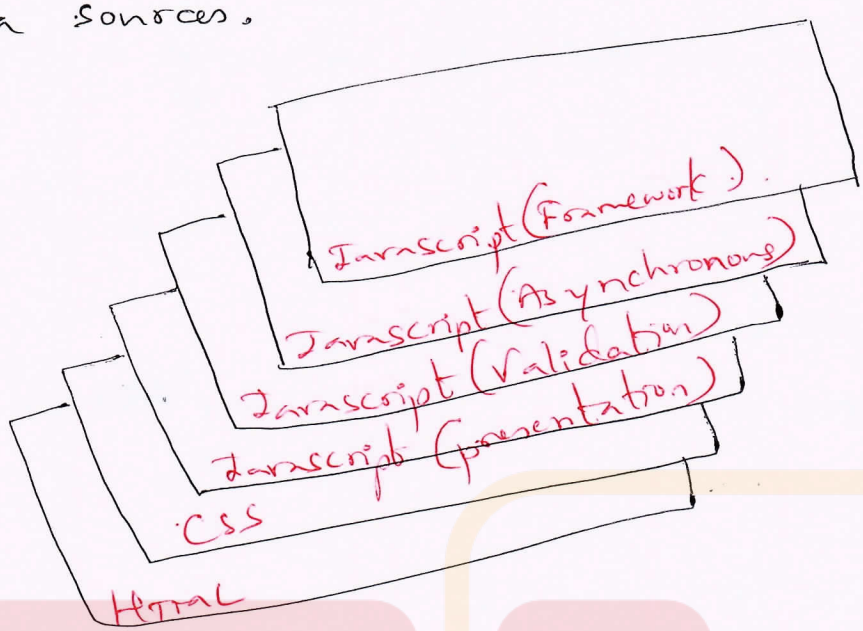
```
footer { background-color : #796d6d;
```

5.a) Software layer is a way of conceptually grouping programming classes which has similar functionality & dependencies.

Common s/w design layer names are :-

* Presentation layer :-
  classes focused on the user interface. It includes creating, hiding & showing divs, using tabs to show multiple views or arrows to page through result sets.
* Business layer :-
  classes that model real world entities such as customers, products & sales.

* Data layer :-
Classes that handle the interaction with data sources. [8M]



Javascript layers.

b) Form events are the main which user input is collected & transmitted to the server. [05M]
The form Events are :- onblur, onchange, onfocus, onreset, onselect & onSubmit.

Onblur :- Form element has lost focus.
Onchange :- some <input>, <textarea> or <select> field had their value change.
onfocus :- Complementing onblur event, this event is triggered to get focus.
onreset :- HTML forms have ability to be reset.
onselect :- When user selects some text.
onsubmit :- When form is submitted this event is triggered.

c) There are 3 different ways to link ~~JS~~ to HTML.
*Inline :- includes JS within certain HTML attributes.
eg:- More
<input type = "button" onclick = "alert("Are you sure?")" />

*Embedded JS :- It refers to placing of JS code within <script> element. Like inline JS, embedded scripts can be difficult to maintain. [3M]

eg:- <script type="text/javascript">
alert("Hello Friends!!!");
</script>

* External JS :- Externally we can include the JS code in HTML. By convention, JS external files have extensions .js.

eg:- <head> <script type="text/javascript" src="greet.js"> </script> </head>

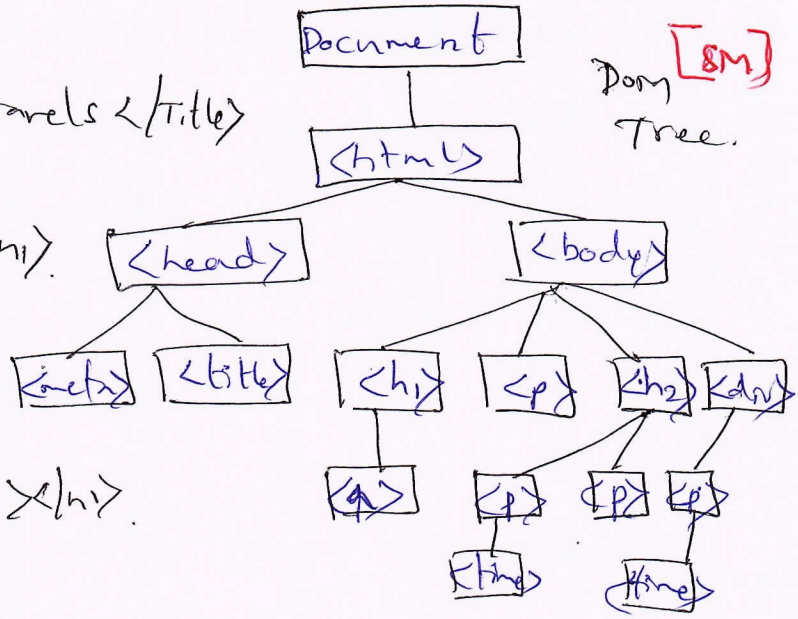
6.a). The approach of adding functional replacements for the web application without Javascript is also referred to as FAIL-SAFE DESIGN : [4M]

*It is a phrase with a meaning beyond the web development. It means when a plan fails then the system's design will still work properly.

b). The different way of programmatically accessing the elements & attributes within the HTML. This is accomplished through a programming interface called DOCUMENT OBJECT MODEL (DOM).

eg:- <html> <head>
<meta> <title> Travels </Title>
</head>
<body> <h1> Your Travels </h1>
 photo ---

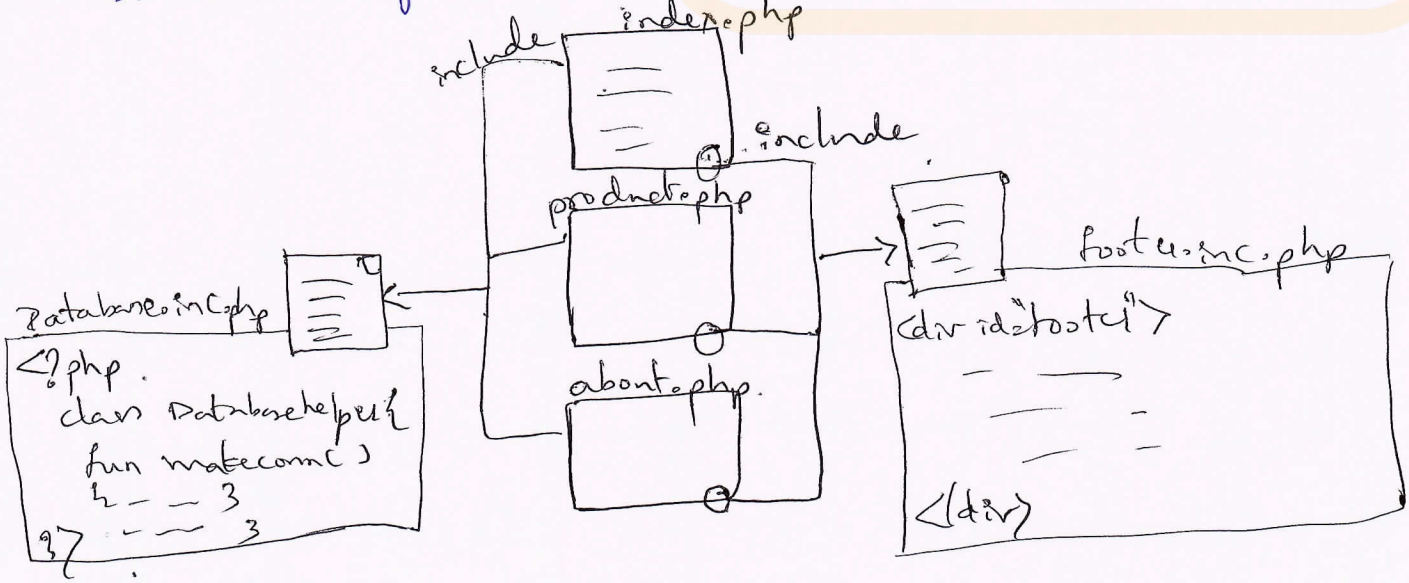
<img src="pak.jpg">
<h2> Reviews --- </h1>
  : - -
 - - -
 </body> </html>



Dom Tree. [SM]

c). Server-side include files are type of encapsulation, nothing but it is like copying & pasting by the selves.

* We can embedd different types of files within the web application. [SM]
* PHP does have one imp facility to insert or include content from one file into another.
* Include files provide a mechanism for reusing both markup & PHP code.



* If the included file doesn't exist or server doesn't have the permission to access, then the include files display an error msg & stops execution.
* The process will not be halted. So it continues the execution of web application.

7a) PHP uses special predefined associative arrays called Superglobal Arrays, which allow the programmer to easily access HTTP headers, query string parameters & commonly needed information. [10m]

* The Superglobal Arrays are :-
  => $_GLOBAL, $_COOKIES, $_ENV, $_FILES, $_GET, $_POST, $_REQUEST, $_SESSION, $_SERVER
* The $_GET & $_POST methods are used to determine the value sent via Query String.
* isset() function is used to determine the value sent via query string.

eg:-
```
<?php
if($_SERVER["REQUEST_METHOD"] == "POST") {
    if(isset($_POST["name"])
    {
        echo "User login Details: ";
        echo "<br> — here we could redirect or authenticate";
        echo "<br> & hide login form or something else";
    }
}
?>
```

b) There are two basic techniques for read/writing files in Php.

* Stream Access :- Read just a small portion of file at a time. It is most-memory-efficient approach when reading very large files.
* The function fopen(), fclose() & fgets() are used to access the content of file.
* To write data to file by using fwrite() in which the same as fgets() passing the file handle & string to write.

eg:-
```
$f = fopen("sample.txt", "r");
$ln = 0;
while($line = fgets($f)) {
    $ln++;
    printf("%d", $ln);
    echo $line . "<br>";
}
fclose($f);
```

* In-memory File Access :- It is simple & easiest way to use to read/write file. There are 3 functions.

⇒ file() - Reads entire file into an array
⇒ file_get_contents() ⇒ Reads " " a string valiable
⇒ file_put_contents() - write contents of a string " out to a file.

eg:-
```
$fileStr = file_get_contents(FILENAME);
// Read contents of file
file_put_contents(FILENAME, $fname); // write contents of file.
```

8.a) Class which consists of properties & methods. Each property in class using different accss Specifiers.

Eg:- class Artist { public firstname;
public Lastname;
getname();
}

* Object is instatiation of a class. to make use of properties & methods of a class using new keyword.

$a = new Artist();

* DATA ENCAPSULATION :-

which refers to restricting accss to an object internal components. There are 2 methods for accsing & modifying properties : Getter & Setters.

public function getName() { return $this->firstname; }

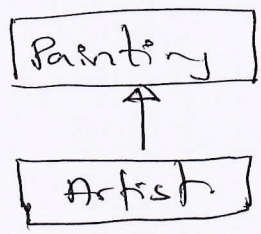
public function setName($n)
{ $this->firstName = $n; }

* INHERITANCE :-

A class inherit from another class is called subclass or derived class. It inherits all of its public & protected methods & properties.

eg:- class Painting extends Artist { ---- }

$p = new Painting.
echo $p->getName();
echo $p->setName("foo");



* POLYMORPHISM :- The method have many forms of its definition. An object can be multiple things at the same time. [10M]

```
$p = new Artist ("Pablo");
$q = new Painting ("1937", $p); $r=new Sculpture("Chicago");
$p -> addwork($q);
$p -> addwork ($r);
```

*Object Interface :- It is way of defining a formal list of methods that a class must implement without specifying implementation.

```
interface viewable {
    public function getSize();
    public function getPNG();
}
```

* Interface contains no properties & its methods do not have method bodies defined.

b) Constructors which specify parameters during instantiation to initialize properties within a class. [10M]

*In PHP constructors are defined as function with the name --construct().

* In the constructor each parameter is assigned to an internal class variable using $this->pro_name.

```
class Artist {
    function __construct($fname, $lname, $city)
    { $this->firstName = $fname;   $this->$city = $city;
    } $this->lastName = $lname;
```

```
$p = new Artist("Foo", "Shaibh", "Gulbarga");
$d = new Artist("Mohit", "JAAN", "Mumbai");
```

* Destructor :-

* It is called when object is destructed or script is stopped or exited. Whenever we create a --destruct PHP will automatically call this function at the end of script.

eg:-
```
public function __destruct()
{
    echo "Artist name: ". $this->name;
}
```

q.a) HTTP Cookies are a client-side approach for persisting state information. The cookie information is stored and retrieved by browser information in a cookie travel within the HTTP header. [8M]

* HTTP Cookies are handled in PHP by two ways :-

⇒ Browser will delete a cookie which are beyond their expiry date.

⇒ If the cookie doesn't have a expiry date specified the browser will delete it when the browser closes.

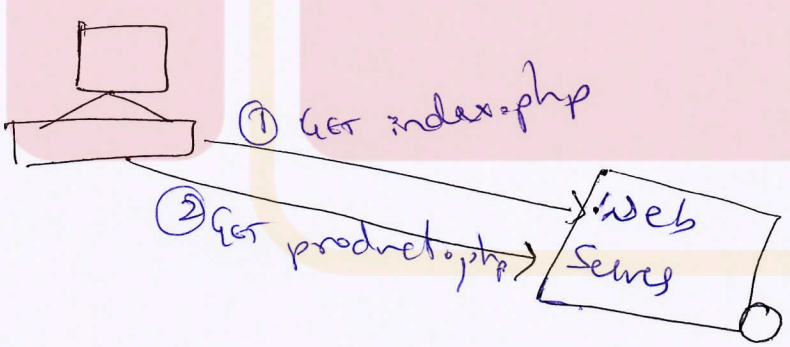
* Cookies can be created & retrieved using setcookie() function.

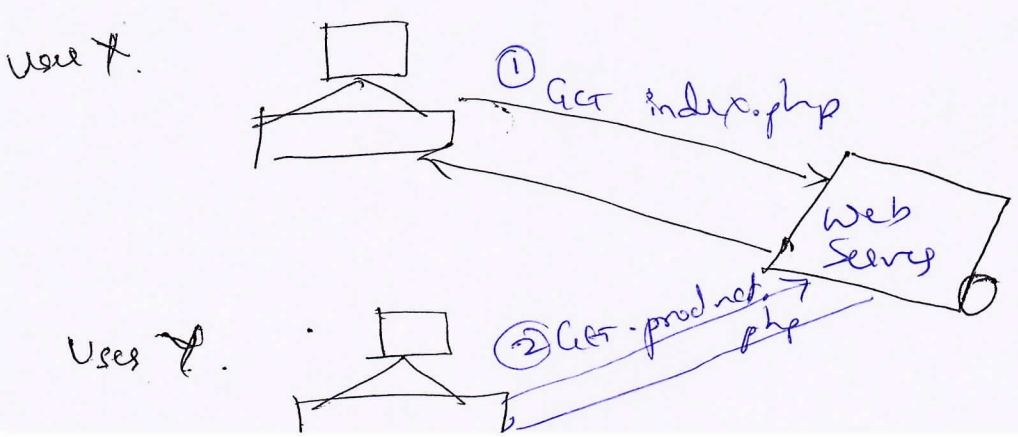
*. If cookie has expired then client's browser would not send anything.

9.b) State is a problem for web application because of sharing information with another request/user:

* All web applications need to be processed with user inputs, output information & read&write from databases or other storage media.
* Single user - desktop applications do not have problem to access the information stored in memory.
* Web application consists of series of disconnected HTTP requests to a web server where each request for a server page is request to run separate program.
* HTTP protocol doesn't distinguish two requests by one src from two requests from two different sources.

*





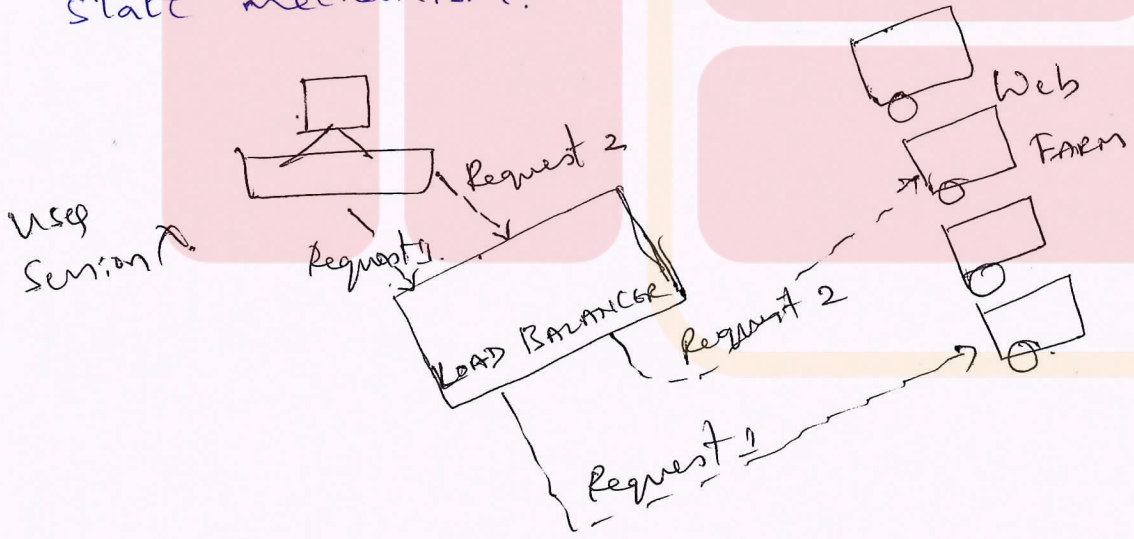
c) $() → can be confusing to PHP developers at first because the $ is used for variable declaration. [4M]

* It selects DOM object which match CSS attributes.
* Pass in the string of CSS selector to $() & result will be set of DOM objects matching the selectors.
* Eg:- $("*"), $("tag"), $(".class"), $("#id");

[5M]

10. a) The sessions are stored between requests in 2 different ways:-

* Configuring the load balancer to be "Session aware" & relate all requests using session to same server.
* Use a shared location to store sessions either in a database, memcache or some other shared session state mechanism.



b) jquery to select all the <P> that contains word "Hello". [5M]

```
var all = $("p *:contains('Hello')");
```

c) Animation used commonly in jQuery are :-

* hide() & show().
* fadeIn & fadeOut()
* SlideUp() & SlideDown()
* Toggel.

* hide() & Show() methods can be called with no arguments to perform a default animation. The new version takes 2 parameters : duration & callback method to execute on completion.

* FadeIn() & FadeOut() :- which used to control the opacity of an element. It takes two parameters as duration & callback method.

* SlideUp/SlideDown() :- These method don't touch the opacity of an element rather gradually change its height.

* Toggle Methods :- jQuery has 'toggel' method to get the visible & hidden states.

* To gel methods between fading in & fading out use fadeToggel() method, toggling between the two sliding states can be achieved using slideToggel().

eg :-
```
$(this).hide();
$(this).html('<a href="http://google.com">Mail</a>');
$(this).show(1000);
```

Verified Clarke

*****

Dean, Academics.

Prepared By :-

Prof. HARSHA.N.N.