



KTU NOTES

The learning companion.

**KTU STUDY MATERIALS | SYLLABUS | LIVE
NOTIFICATIONS | SOLVED QUESTION PAPERS**

Website: www.ktunotes.in

CST 463 Web Programming Syllabus Model Question Answers

1. Define WWW. List any two examples of web server & web browser. Differentiate between URL and a domain?

WWW stands for World Wide Web, which is a system of interlinked, hypertext documents accessed via the Internet.

Examples of web servers include Apache and Nginx. Examples of web browsers include Google Chrome and Mozilla Firefox.

A URL (Uniform Resource Locator) is the specific address of a webpage or file on the Internet, including the protocol (e.g. http or https) and domain name. For example, "https://www.example.com/page.html" is a URL.

A domain is the part of a URL that specifies the Internet address of a website. In the example above, "example.com" is the domain. A domain name is the human-friendly form of an IP address and is used to identify a website.

2. Write the syntax of the URL? Rewrite the default URL of your university website by adding a subdomain named 'Research' and a web page named 'FAQ.html'. Also link this URL through the logo of 'kturesearch.png' placed in a web page. The FAQ page should be opened in a new window.

The syntax of a URL is as follows:

[protocol]://[subdomain].[domain].[top-leveldomain]/[path/to/resource]

Example of a university website URL:

<https://www.ktu.edu.in/>

To add a subdomain named 'Research' and a web page named 'FAQ.html' to the default URL of the university website, it would be written as follows:

<https://research.ktu.edu.in/FAQ.html>

To link the above URL through the logo of 'kturesearch.png' placed in a web page, the following HTML code could be used:

```
<a href="https://research.exampleuniversity.edu/FAQ.html"
target="_blank">
    
</a>
```

Here the 'target="_blank"' attribute tells the browser to open the link in a new window.

3. Illustrate the implementation of a JavaScript function greeting () using external .js file, to display a welcome message, when you click on a Button in an HTML page.

External .js

```
function greeting() {
    alert("Welcome to our website!");
}
```

index.html

```
<!DOCTYPE html>
<html>
    <head>
        <script type="text/javascript"
src="greeting.js"></script>
    </head>
    <body>
        <button onclick="greeting()">Click me</button>
    </body>
</html>
```

4. What are different ways of adjusting spacing in a text with suitable example.

There are several ways to adjust spacing in text, including:

Line spacing: This refers to the space between lines of text. It can be adjusted using the CSS property "line-height".

```
p {  
    line-height: 1.5;  
}
```

In this example, the line height of all text within a "p" element will be set to 1.5, resulting in a space that is 1.5 times the size of the font.

Letter spacing: This refers to the space between characters of text. It can be adjusted using the CSS property "letter-spacing".

```
h1 {  
    letter-spacing: 2px;  
}
```

In this example, the letter spacing of all text within an "h1" element will be increased by 2 pixels.

Word spacing: This refers to the space between words. It can be adjusted using the CSS property "word-spacing".

```
span {  
    word-spacing: 20px;  
}
```

In this example, the word spacing of all text within a "span" element will be increased by 20 pixels.

5. Discuss the various CSS style sheet levels with suitable examples. How are conflicts resolved when multiple style rules apply to a single web page element?

There are three levels of CSS style sheets:

Inline styles: These styles are applied directly to an HTML element using the "style" attribute. For example: `<p style="color: red;">This is a red paragraph.</p>`

Internal styles: These styles are included within the `<head>` section of an HTML document using the `<style>` tag. For example:

```
<head>
  <style>
    p {
      color: red;
    }
  </style>
</head>
<body>
  <p>This is a red paragraph.</p>
</body>
```

External styles: These styles are stored in a separate CSS file, which is linked to an HTML document using the `<link>` tag. For example:

```
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <p>This is a red paragraph.</p>
</body>
```

CSS styles are applied to elements based on specificity, which means that the browser follows a set of rules to determine which style should be applied to an element when multiple styles apply to it. Specificity is calculated by assigning a weight to each selector, and the style with the highest weight is applied to the element.

For example, if an element has two conflicting styles, one with an ID selector (#example) and one with a class selector (.example), the ID selector will have a higher specificity and its styles will be applied to the element.

When two styles have the same specificity, the one that appears later in the CSS file will take precedence.

6. Describe how input from an HTML form is retrieved in a PHP program, with an example

In an HTML form, input is collected in various form elements, such as text fields, checkboxes, and radio buttons. When the user submits the form, the data is sent to a server-side script, such as a PHP program, for processing.

In PHP, the data from the form can be retrieved using the \$_POST or \$_GET superglobal arrays. The \$_POST array is used for data submitted via the HTTP POST method, while the \$_GET array is used for data submitted via the HTTP GET method. The method is specified in the form tag.

For example, consider the following HTML form:

```
<form method="post" action="submit.php">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name">
  <br>
  <label for="email">Email:</label>
  <input type="email" id="email" name="email">
  <br>
  <input type="submit" value="Submit">
</form>
```

In the PHP script "submit.php", the data from the form can be retrieved like this:

```
$name = $_POST['name'];  
$email = $_POST['email'];
```

You can also use a function `filter_input()` to retrieve the form data, which provides an extra layer of security by validating and sanitizing the input.

```
$name = filter_input(INPUT_POST, 'name', FILTER_SANITIZE_STRING);  
$email = filter_input(INPUT_POST, 'email', FILTER_VALIDATE_EMAIL);
```

This way you can retrieve the form data, process it and store it in a database or use it to generate a response to the user, such as sending an email or displaying a message on the page.

7. Write a PHP program to check whether a number is prime number or not.

```
<?php  
function checkPrime($num) {  
    if ($num == 1) {  
        return false;  
    }  
    for ($i = 2; $i <= sqrt($num); $i++) {  
        if ($num % $i == 0) {  
            return false;  
        }  
    }  
    return true;  
}  
  
$num = 7;
```

```

    if (checkPrime($num)) {
        echo $num . " is a prime number.";
    } else {
        echo $num . " is not a prime number.";
    }
?>

```

8. Discuss the various steps for establishing PHP-MySQL connection with a MySQL

- Install PHP MySQL extension
- Create a new MySQL user and password
- Use `mysqli_connect()` function to connect to the MySQL server
- Test the connection using `mysqli_connect_errno()` and `mysqli_connect_error()`
- Close the connection using `mysqli_close()`
- Use try-catch block to handle errors
- Consider using PDO for more secure and flexible connection.

9. Describe the schema of a document implemented in JSON with suitable examples

A JSON document schema is a blueprint that defines the structure and format of a JSON document. It includes information about the data types, fields, and validation rules for the data within the document.

Example:

```

{
  "type": "object",
  "properties": {
    "name": {
      "type": "string"
    },
    "age": {
      "type": "integer"
    }
  }
}

```



```

    },
    "address": {
      "type": "object",
      "properties": {
        "street": {
          "type": "string"
        },
        "city": {
          "type": "string"
        },
        "state": {
          "type": "string"
        }
      },
      "required": [
        "street",
        "city",
        "state"
      ]
    },
    "required": [
      "name",
      "age",
      "address"
    ]
  }
}

```

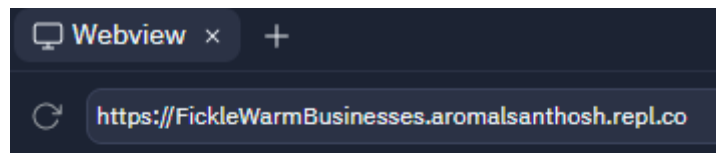
This schema defines a JSON document with an object that has a "name" field of type string, an "age" field of type integer, and an "address" field that is also an object. The "address" object has three fields: "street", "city", and "state" all of which are of type string. Additionally, it specifies that the "name", "age", and "address" fields are required in the document.

10. Explain the role of Resource controllers in Laravel.

In Laravel, a resource controller is a controller class that is used to handle all HTTP requests for a specific resource. These controllers are designed to handle the basic CRUD (create, read, update, and delete) operations for a resource, and

are typically used to handle requests for a resource that maps to a database table. Resource controllers include methods for handling requests for creating new resources, displaying existing resources, updating existing resources, and deleting resources. They are typically defined using a set of routes that map to the controller's methods, and can be easily generated using the command-line interface (CLI) tool included with Laravel.

11. (a) Design a webpage that displays the following table.



Food Item	Recommended Intake			
	age < 15		age > 15	
	gm	Kcal	gm	Kcal
Cerials	1000	2000	750	1760
Non-Cerials	450	800	350	600

Ktunotes.in

```

<html>
<style>
table, th, td {
    border:1px solid black;
}
</style>
<body>
<table>
    <tr>
        <th colspan="1" rowspan="3">Food Item</th>
        <th colspan="4">Recommended Intake</th>
    </tr>
    <tr>
        <th colspan="2">age < 15</th>
        <th colspan="2">age > 15</th>
    </tr>
    <tr>
        <td>gm</td>
        <td>Kcal</td>
        <td>gm</td>
        <td>Kcal</td>
    </tr>
    <tr>
        <th>Cerials</th>
        <td>1000</td>
        <td>2000</td>
        <td>750</td>
        <td>1760</td>
    </tr>
    <tr>
        <th>Non-Cerials</th>
        <td>450</td>
        <td>800</td>
        <td>350</td>
        <td>600</td>
    </tr>
</table>
</body>
</html>

```

(b) What is the difference between radio buttons and checkboxes when implemented using HTML? Write HTML code to implement a form which has the following elements:

- A textbox which can accept a maximum of 25 characters
- Three radio buttons with valid Label, Names and values
- Three check boxes buttons with valid Label, Names and values
- A selection list containing four items, two which are always visible
- A submit button clicking on which will prompt the browser to send the form data to the server "http://www..mysite.com/reg.php" using "POST" method and reset button to clear its contents. You can use any text of your choice to label the form elements.

The main difference between radio buttons and checkboxes when implemented using HTML is that radio buttons are used to select one option from multiple options, whereas checkboxes are used to select multiple options from multiple options.

Here is an example of how you might implement a form that has the elements you described using HTML:

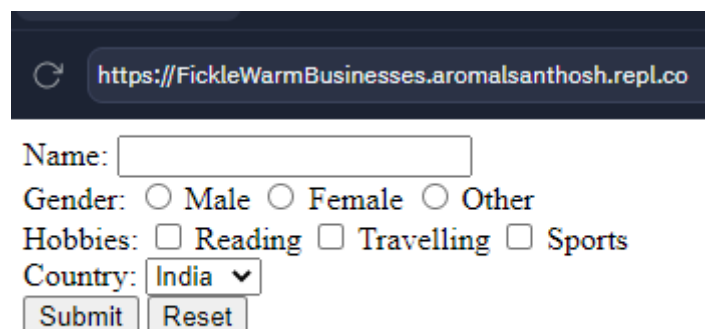
```
<form action="http://www.mysite.com/reg.php"
method="POST">
  <label for="name">Name:</label>
  <input type="text" id="name" name="name"
maxlength="25">
  <br>
  <label for="gender">Gender:</label>
  <input type="radio" id="male" name="gender"
value="male">
  <label for="male">Male</label>
  <input type="radio" id="female" name="gender"
value="female">
  <label for="female">Female</label>
  <input type="radio" id="other" name="gender"
value="other">
```

```

<label for="other">Other</label>
<br>
<label for="hobbies">Hobbies:</label>
<input type="checkbox" id="reading" name="hobbies[]"
value="reading">
  <label for="reading">Reading</label>
  <input type="checkbox" id="travelling" name="hobbies[]"
value="travelling">
    <label for="travelling">Travelling</label>
    <input type="checkbox" id="sports" name="hobbies[]"
value="sports">
      <label for="sports">Sports</label>
    <br>
  <label for="country">Country:</label>
  <select id="country" name="country">
    <option value="india">India</option>
    <option value="usa">USA</option>
    <option value="uk">UK</option>
    <option value="other">Other</option>
  </select>
  <br>
  <input type="submit" value="Submit">
  <input type="reset" value="Reset">
</form>

```

OUTPUT :



Name:
 Gender: ☐ Male ☐ Female ☐ Other
 Hobbies: ☐ Reading ☐ Travelling ☐ Sports
 Country:

In the above code:

The textbox is created using the `<input type="text">` element.

The radio buttons are created using the `<input type="radio">` element, with each one having a unique "id" and "name" attribute, and a common "name" attribute.

The checkboxes are created using the `<input type="checkbox">` element, with each one having a unique "id" and "name" attribute, and a common "name" attribute.

The selection list is created using the `<select>` and `<option>` elements, with the `<select>` element having an "id" and "name" attribute, and the `<option>` elements having "value" attribute.

The submit button is created using the `<input type="submit">` element, and the reset button is created using the `<input type="reset">` element.

It's important to note that the "name" attribute is used to identify the field when the form is submitted, and the "id" attribute is used to associate the label element with the field.

12. (a) Write the equivalent HTML code to implement the following in a web page:

(i) An image titled "birds.jpg" with a height of 100 pixels and width of 200 pixels. If the image cannot be accessed, a message "No image available" should be displayed

```
<!DOCTYPE html>
<html>
<body>
    
</body>
</html>
```

(ii) A hyperlink to the URL

"www.mysite.com/birds.jpg". The hyperlink should have the label "Click

Here".

```
<!DOCTYPE html>
<html>
<body>
    <a href="http://www.mysite.com/birds.jpg">Click
Here</a>
</body>
</html>
```

(b) Create a static HTML document for your portfolio, which includes the following contents: your name, address, Mobile Number and email address. Also add the details about your college, university, your major and the batch of study. Include a picture of yourself and at least one other image (friend/pet/role model) to the document with a short description about that. Add three paragraphs about your personal history, with links to your social media profile. Also create an ordered list for describing your Skill Set & an unordered list showing your Strengths & Weaknesses.

```
<!DOCTYPE html>
<html>

<head>
    <title>My Portfolio</title>
</head>

<body>
    <h1>My Portfolio</h1>
    <div>
        
        <h2>Personal Information</h2>
        <p>Name: John Doe</p>
        <p>Address: 123 Main St, Anytown USA</p>
        <p>Mobile Number: 555-555-5555</p>
```

```
<p>Email: johndoe@email.com</p>
<h2>Education</h2>
<p>College: XYZ University</p>
<p>Major: Computer Science</p>
<p>Batch: 2012-2016</p>
</div>
<div>
  
  <h2>About the image</h2>
  <p>This is a picture of my friend, who has always
been a source of inspiration for me. He is a successful
entrepreneur and has taught me a lot about perseverance
and hard work.</p>
</div>
<div>
  <h2>Personal History</h2>
  <p>I was born and raised in Anytown USA. I have
always had a passion for technology and coding. I studied
Computer Science in college and have been working as a
software engineer for the past 5 years. You can find more
information about me on my social media profiles:</p>
  <ul>
    <li><a
href="https://www.linkedin.com/in/johndoe">LinkedIn</a></li>
    <li><a
href="https://www.facebook.com/johndoe">Facebook</a></li>
    <li><a
href="https://twitter.com/johndoe">Twitter</a></li>
  </ul>
  <h2>Skills</h2>
  <ol>
    <li>JavaScript</li>
    <li>ReactJS</li>
    <li>Node.js</li>
    <li>Python</li>
  </ol>
</div>
```



```

    <li>SQL</li>
</ol>
<h2>Strengths and Weaknesses</h2>
<ul>
  <li>Strengths:</li>
    <ul>
      <li> Strong problem solving skills</li>
      <li> Good at working under pressure</li>
      <li> Strong attention to detail</li>
    </ul>
  <li>Weaknesses:</li>
    <ul>
      <li> Tendency to be a perfectionist</li>
      <li> Poor time management</li>
      <li> Tendency to procrastinate</li>
    </ul>
  </li>
</ul>
</div>
</body>

</html>

```

13. (a) Illustrate the usage of JavaScript DOM in event handling and explain any three methods With example.

JavaScript DOM (Document Object Model) allows you to access and manipulate elements in an HTML or XML document. One way to use the DOM in JavaScript is through event handling. This allows you to specify what happens when a user interacts with an element on the page, such as clicking a button or hovering over a link.

Here are three examples of DOM event handling methods in JavaScript:

addEventListener: This method allows you to attach an event listener to a specific element, so that a function is called whenever the event occurs. For example:

```
document.getElementById("myButton").addEventListener("click", function(){
    alert("Button was clicked!");
});
```

removeEventListener: This method allows you to remove an event listener that was previously added to an element. For example:

```
var button = document.getElementById("myButton");

// Add an event listener
button.addEventListener("click", function(){
    alert("Button was clicked!");
});

// Remove the event listener
button.removeEventListener("click", function(){
    alert("Button was clicked!");
});
```

onclick: This is a property of an element that allows you to specify a function to be called when the element is clicked. For example:

```
<button id="myButton" onclick="alert('Button was
clicked!')">Click me</button>
```

This is a short way of doing the same thing as the first example, but the function is defined directly on the element rather than being added as an event listener.

(b) Write CSS and the corresponding HTML code for the following:

- Set the background color for the hover and active link states to "green"
- Set the list style for unordered lists to "square".
- Set "Flower.png as the background image Of the page and set 3% margin for the pages
- Set dashed border for left and right and double border for top & bottom
- of a table with 2 rows.

HTML :

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/css"
href="styles.css">
  </head>
  <body>
    <ul>
      <li>List item 1</li>
      <li>List item 2</li>
      <li>List item 3</li>
    </ul>
    <table>
      <tr>
        <td>Table cell 1</td>
        <td>Table cell 2</td>
      </tr>
      <tr>
        <td>Table cell 3</td>
        <td>Table cell 4</td>
      </tr>
    </table>
  </body>
</html>
```

CSS :

```
a:hover, a:active {  
    background-color: green;  
}
```

```
ul {  
    list-style-type: square;  
}
```

```
body {  
    background-image: url("Flower.png");  
    margin: 3%;  
}
```

```
table {  
    border-left: dashed;  
    border-right: dashed;  
    border-top: double;  
    border-bottom: double;  
}
```

14. (a) List the order of precedence of style levels. Organize a sample web page for

providing *KTU BTech Honours Regulation 19' for KTU and use embedded

Style sheet to apply minimum 5 styles for list, tables and pages.

The order of precedence of style levels is as follows:

1. Inline styles (applied directly to an HTML element)
2. Internal styles (defined within the <style> tag of an HTML document)
3. External styles (defined in a separate .css file and linked to an HTML document)

```
<!DOCTYPE html>  
<html>  
<head>
```

```

<title>KTU BTech Honours Regulation 19</title>
<style>
  /* Embedded styles */
  h1 {
    font-size: 24px;
  }
  ul {
    list-style-type: square;
  }
  table {
    border-collapse: collapse;
  }
  th, td {
    padding: 8px;
    text-align: left;
  }
  /* Additional styles */
  .highlight {
    background-color: #f2f2f2;
  }
</style>
</head>
<body>
  <h1>KTU BTech Honours Regulation 19</h1>
  <ul>
    <li>Eligibility for admission</li>
    <li>Curriculum and syllabus</li>
    <li>Examination and evaluation</li>
  </ul>
  <table>
    <tr>
      <th>Semester</th>
      <th>Subjects</th>
    </tr>
    <tr class="highlight">
      <td>1</td>

```

```
        <td>Mathematics, Physics, Chemistry</td>
    </tr>
    <tr>
        <td>2</td>
        <td>Computer Programming, Electrical Engineering,
Mechanics</td>
    </tr>
</table>
</body>
</html>
```

(b) Illustrate the different ways of Array declaration in JavaScript. Describe the function of the following JavaScript Array object methods with examples.
(i) join (ii) slice

There are several ways to declare an array in JavaScript:

Using the Array constructor:

```
var arr1 = new Array();
var arr2 = new Array(1, 2, 3);
var arr3 = new Array(5); // Creates an array with 5 empty
slots
```

Using array literals:

```
var arr1 = [];
var arr2 = [1, 2, 3];
```

Using the spread operator:

```
var arr1 = [...[1, 2, 3]];
var arr2 = [...new Array(5)]; // Creates an array with 5
undefined elements
```

Using Array.of() method

```
var arr1 = Array.of(1, 2, 3);
var arr2 = Array.of(5); // Creates an array with one
element with the value 5
```

join() method is used to join all the elements of an array into a string. The elements are separated by a specified separator.

Example:

```
var arr = ["apple", "banana", "orange"];
console.log(arr.join());    // "apple,banana,orange"
console.log(arr.join(" ")); // "apple banana orange"
console.log(arr.join("-")); // "apple-banana-orange"
```

slice() method is used to extract a section of an array and return a new array. This method takes two arguments: the starting index and the ending index (not included). If the ending index is not provided, the slice will include all the elements from the starting index to the end of the array.

Example:

```
var arr = [1, 2, 3, 4, 5];
console.log(arr.slice(1));    // [2, 3, 4, 5]
console.log(arr.slice(1, 3)); // [2, 3]
```

15. (a) Explain any six string handling functions used in PHP with example.

1. `strlen()` - This function returns the length of a string. For example, if we have a string "Hello World!", the `strlen()` function would return 12.

```
$string = "Hello World!"; echo strlen($string); // Output: 12
```

2. `strpos()` - This function searches for a specific text within a string and returns its position. If the text is not found, it returns false.

```
$string = "Hello World!"; echo strpos($string, "World"); //  
Output: 6
```

3. `str_replace()` - This function replaces all occurrences of a search text with a replacement text in a given string.

```
$string = "Hello World!";  
$new_string = str_replace("World", "PHP", $string);  
echo $new_string; // Output: "Hello PHP!"
```

4. `substr()` - This function returns a portion of a string. You can specify the starting position and the length of the substring.

```
$string = "Hello World!";  
echo substr($string, 0, 5); // Output: "Hello"
```


5. strtolower() - This function converts all characters in a string to lowercase.

```
$string = "Hello World!";  
echo strtolower($string); // Output: "hello world!"
```

6. strtoupper() - This function converts all characters in a string to uppercase.

```
$string = "Hello World!";  
echo strtoupper($string); // Output: "HELLO WORLD!"
```

(b) How does a PHP array differ from an array in C? List the different ways to create an array in PHP with an example, Explain any 4 functions that deals with a PHP array.

PHP arrays and C arrays are similar in that they both allow for the storage of multiple values in a single variable. However, there are a few key differences between the two:

PHP arrays are dynamic in size, meaning that their size can be changed at runtime, whereas C arrays have a fixed size that is determined at compile-time.

PHP arrays can store values of different data types, whereas C arrays can only store values of a single data type.

PHP arrays use associative keys to access elements, whereas C arrays use integer indices.

There are several ways to create an array in PHP:

Using the array() function:

```
$my_array = array("apple", "banana", "orange");
```

Using square brackets:

```
$my_array = ["apple", "banana", "orange"];
```

Using the short array syntax:

```
$my_array = ["apple", "banana", "orange"];
```

Using the array literal notation:

```
$my_array = [  
    "apple",  
    "banana",  
    "orange"  
];
```

count() - This function returns the number of elements in an array.

```
$my_array = array("apple", "banana", "orange");  
echo count($my_array); // Output: 3
```

array_search() - This function searches for a value in an array and returns its key if found, and false otherwise.

```
$my_array = array("apple", "banana", "orange");  
echo array_search("banana", $my_array); // Output: 1
```

sort() - This function sorts the elements of an array in ascending order.

```
$my_array = array("apple", "banana", "orange");
```

```
sort($my_array);  
print_r($my_array); // Output: Array ( [0] => apple [1]  
=> banana [2] => orange )
```

array_slice() - This function returns selected parts of an array.

```
$my_array = array("apple", "banana", "orange");  
print_r(array_slice($my_array, 1, 2)); // Output: Array (  
[0] => banana [1] => orange )
```

16. (a) During the process of fetching a web page from a web server to a client

browser, at what point does an embedded PHP script get executed. What are

the two modes that the PHP processor operates in? Explain

During the process of fetching a web page from a web server to a client browser, an embedded PHP script gets executed on the server side before the web page is sent to the client browser.

When a client browser requests a web page, the web server receives the request and checks if the requested page contains any PHP code. If it does, the web server passes the PHP code to the PHP processor for execution. The PHP processor then executes the code and generates the HTML, CSS, and JavaScript that make up the web page. This output is then sent back to the web server, which in turn sends it to the client browser as a response to the original request.

The PHP processor operates in two modes:

1. The first mode is called the "Web Server mode". In this mode, the PHP code is executed as a part of a web server process. The PHP code is executed on the server side and the output is sent to the client browser as HTML, CSS, and JavaScript. This mode is typically used to generate dynamic web pages that are tailored to the specific needs of each user.

2. The second mode is called the "Command Line mode". In this mode, the PHP code is executed as a command-line script, rather than as part of a web server process. This mode is typically used for running automated scripts and tasks, such as sending emails or generating reports. The output is not sent to a web browser, instead it can be sent to a file or to the command line interface.

In the web server mode, PHP code is executed by the PHP engine, which is built into the web server. The PHP code is parsed, executed, and the output is returned to the web server, which in turn sends it to the client browser.

In the command line mode, PHP code is executed by the PHP command-line interpreter, which runs independently of the web server. The PHP code is parsed and executed, and the output is returned to the command-line interface.

(b) Why is PHP considered to be dynamically typed? Distinguish between implode and explode function in PHP with suitable examples.

PHP is considered to be dynamically typed because variables in PHP do not have to be declared with a specific data type. Instead, the data type of a variable is determined automatically based on the value that is assigned to it. This means that the same variable can be used to store different types of data (e.g. a string, an integer, or an array) at different points in the script without the need to redeclare it.

The implode() function in PHP is used to join an array of strings into a single string, using a specified delimiter. For example, consider the following array:

```
$array = array("apple", "banana", "orange");
```

We can use the implode() function to join this array into a single string, with the delimiter being a comma:

```
$string = implode(",", $array);
```

This will result in the string "apple,banana,orange".

On the other hand, the explode() function in PHP is used to split a string into an array, using a specified delimiter. For example, consider the following string:

```
$string = "apple,banana,orange";
```

We can use the explode() function to split this string into an array, with the delimiter being a comma:

```
$array = explode(",", $string);
```

This will result in the array ["apple", "banana", "orange"].

In summary, the implode() function is used to join an array into a single string, while the explode() function is used to split a string into an array. Both functions use a specified delimiter to determine where to split or join the elements in the array or string.

17. (a) Write equivalent PHP statements corresponding to the following:

- Declare an associative array named "ages" to store the key-value pairs ("Alice", 30), ("Bob", 30), ("Harry", 35), ("Mary", 32).
- Modify the value associated With the key "Mary" to 28.
- Sort the array according to values maintaining the key-value relationships and print the sorted key-value pairs.
- The entry identified b the k "Bob"

```
$ages = array("Alice"=>30, "Bob"=>30, "Harry"=>35,
"Mary"=>32);
$ages["Mary"] = 28;
asort($ages);
foreach($ages as $name => $age) {
    echo $name . " is " . $age . " years old. <br>";
}
echo "Bob is " . $ages["Bob"] . " years old.";
```

(b) What are the uses Of cookies in web pages? Describe syntax for setting cookies in PHP. How can you access and delete the cookie using setcookie() function?

Cookies are small text files that are stored on a user's computer by a website. They are used to remember information about the user, such as their preferences or login credentials. Some common uses of cookies include:

- Storing user preferences: Websites can use cookies to remember a user's preferred language, font size, or other settings.
- Keeping a user logged in: A website can use cookies to remember a user's login credentials so that they don't have to enter them every time they visit the site.
- Tracking user behavior: Websites can use cookies to track a user's behavior on the site, such as which pages they visit and how long they stay on the site.

In PHP, cookies are set using the setcookie() function. The syntax for the setcookie() function is as follows:

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

name: The name of the cookie.

value: The value of the cookie.

expire: The expiration time of the cookie (in seconds).

path: The path on the server where the cookie will be available.

domain: The domain that the cookie is available to.

secure: Indicates whether the cookie should only be transmitted over a secure HTTPS connection.

httponly: Indicates whether the cookie should only be accessible through the HTTP protocol.

To access a cookie, you can use the `$_COOKIE` superglobal variable. For example, to access the value of a cookie named "username", you would use

```
$_COOKIE['username'].
```

To delete a cookie, you can use the `setcookie()` function and set the expiration time to a past date. For example, to delete a cookie named "username", you would use the following code:

```
setcookie("username", "", time() - 3600);
```

This will set the expiration time of the cookie to one hour ago, effectively deleting it from the user's computer.

18. (a) Write a PHP form handling program to perform the user registration of any website with a minimum of 5 different fields and insert the data into a MySQL table after establishing necessary connections with the DB,

```
<?php
```

```

// Connect to MySQL
$con = mysqli_connect("hostname", "username", "password",
"database_name");

// Check connection
if (mysqli_connect_errno()) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}

// Form handling code
if (isset($_POST['submit'])) {
    $first_name = $_POST['first_name'];
    $last_name = $_POST['last_name'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $age = $_POST['age'];

    // Insert data into MySQL table
    $sql = "INSERT INTO users (first_name, last_name, email,
password, age)
VALUES ('$first_name', '$last_name', '$email', '$password',
'$age')";

    if (!mysqli_query($con, $sql)) {
        die('Error: ' . mysqli_error($con));
    }
    echo "1 record added";
}
mysqli_close($con);

?>

```

```

<!-- Form HTML code -->
<form method="post" action="">
    <label for="first_name">First Name:</label>
    <input type="text" name="first_name"><br>
    <label for="last_name">Last Name:</label>
    <input type="text" name="last_name"><br>
    <label for="email">Email:</label>
    <input type="email" name="email"><br>
    <label for="password">Password:</label>
    <input type="password" name="password"><br>
    <label for="age">Age:</label>
    <input type="number" name="age"><br>
    <input type="submit" name="submit" value="Submit">

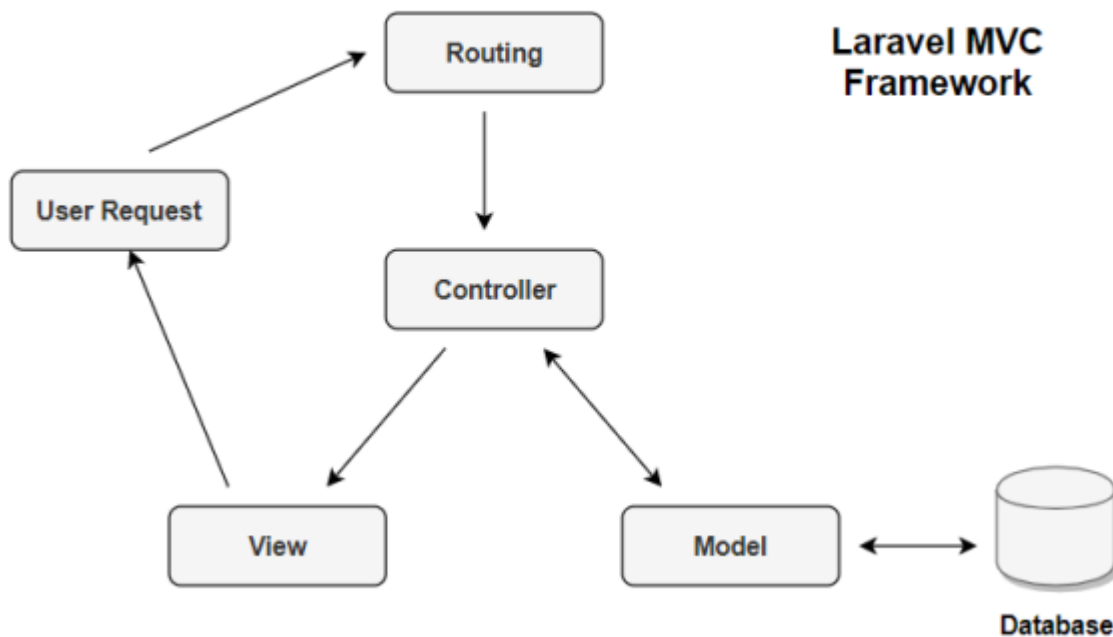
```


</form>

(b) Design the HTML page which enters a given number and embed the PHP code to display a message indicating, whether the number is odd or even, when clicking on the 'CHECK NUMBER' button.

```
<!DOCTYPE html>
<html>
<body>
    <h1>Odd or Even Checker</h1>
    <form action="" method="post">
        <label for="number">Enter a number:</label>
        <input type="text" id="number" name="number">
        <input type="submit" value="CHECK NUMBER" name="submit">
    </form>
    <?php
        if(isset($_POST['submit'])) {
            $number = $_POST['number'];
            if($number % 2 == 0) {
                echo "<p>The number is even.</p>";
            } else {
                echo "<p>The number is odd.</p>";
            }
        }
    ?>
</body>
</html>
```

19. (a) With a neat diagram, explain about Laravel MVC Framework.



Laravel is a well-known PHP web application framework that gives developers access to a variety of tools and functionality for the creation of websites. Taylor Otwell was the one who initially developed it in 2011, and since then it has grown to become one of the most popular PHP frameworks for developing web applications.

It is much simpler for programmers to create and maintain applications that make use of Laravel's elegant syntax and clean codebase, which is one of the framework's most distinguishing characteristics. It uses the Model-View-Controller (MVC) architectural pattern, which helps to isolate the presentation layer from the business logic, making it simpler to design and manage large-scale systems. This pattern also helps to ensure that the presentation layer is always up to date.

In addition, Laravel incorporates a variety of capabilities that are necessary for the building of modern websites. These include database migrations, routing, authentication, and a robust job scheduler. Other features include: In addition to that, it comes with an object-relational mapper (ORM), which makes it possible for developers to deal with databases in a more effective manner and eliminates the need for them to write difficult SQL queries.

Laravel offers a robust ecosystem of third-party packages and libraries that can be easily added into projects, in addition to these fundamental functionalities.

These core features can be found here. Because of this, it will not be difficult for you to incorporate functionality into your application, such as payment gateways, email services, and real-time communication.

Because of its high level of extensibility, Laravel makes it simple for developers to modify and expand the framework so that it better serves their individual requirements. It comes with a comprehensive collection of testing tools, some of which are known as PHPUnit and Dusk, which makes it simple to check whether or not your application is functioning appropriately and whether or not it is prepared for production.

In general, Laravel is a robust and user-friendly framework that has quickly become a favorite option among web developers all around the world. Because of its streamlined syntax, extensive feature set, and active user community, it is an excellent choice for developing contemporary online apps.

(b) Discuss in detail about Laravel 's Routing mechanisms.

Routing is an important aspect of web development, as it determines how a web application responds to requests from clients. Laravel provides a simple and flexible routing mechanism that allows developers to easily map HTTP verbs and URIs to specific actions in a controller.

In Laravel, routes are defined in the `app/Http/routes.php` file. There are several methods available for defining routes, including the basic `get`, `post`, `put`, `patch`, and `delete` methods. These methods correspond to the HTTP verbs of the same name, and allow developers to specify the HTTP verb that a route should respond to.

For example, to define a route that responds to a GET request, you can use the `get` method like this:

```
Route::get('/', function () {  
    return 'Hello World';  
});
```

This route would respond to a GET request to the root URL of the application and return the string "Hello World".

In addition to basic routes, Laravel also supports route parameters. These allow you to capture values from the URI and pass them to your route's action. For example, you might define a route like this:

```
Route::get('/users/{id}', function ($id) {  
    return 'User '.$id;  
});
```

This route would match any GET request to the /users/{id} URI, where {id} is a placeholder for a value that will be passed to the route's action. For example, a request to /users/123 would return the string "User 123".

In addition to basic routes and route parameters, Laravel also supports named routes. These allow you to give your routes a unique name, which can be used to generate URLs or redirects to the route. This is useful if you need to change the URI of a route, as you can simply update the name of the route rather than updating every place in your application where the route is used.

Laravel's routing mechanism is simple and flexible, making it easy for developers to define and manage routes in their web applications. It provides a range of features and options that allow developers to create complex routing structures to suit their needs.

20. (a) Enumerate the data types in JSON. Illustrate the document definition of a Student document using JSON Schema.

JSON has the following data types:

- String
- Number
- Object
- Array
- Boolean

- Null

JSON schema for student document

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Student",
  "type": "object",
  "properties": {
    "name": {
      "type": "string",
      "minLength": 1
    },
    "age": {
      "type": "integer",
      "minimum": 0
    },
    "gender": {
      "type": "string",
      "enum": [ "male", "female" ]
    },
    "gpa": {
      "type": "number",
      "minimum": 0,
      "maximum": 4
    },
    "courses": {
      "type": "array",
      "items": {
        "type": "string"
      }
    }
  },
  "required": [ "name", "age", "gender", "gpa", "courses" ]
}
```

(b) Discuss the following in Laravel Views

- Creating & Rendering Views
- Passing Data to Views
- Sharing Data with All Views

Laravel views are used to display the user interface of a web application. They are typically stored in the "resources/views" directory of a Laravel project.

Creating and rendering views in Laravel is done using the "view" helper function. The function takes the name of the view as its first argument, and an array of data as its second argument. For example, to render a view called "welcome" and pass an array of data to it, you would use the following code:

```
return view('welcome', ['name' => 'John Doe']);
```

Passing data to views is done by including an array of data as the second argument of the "view" function. This data can then be accessed within the view using blade syntax, for example:

```
<h1>Hello, {{ $name }}</h1>
```

Sharing data with all views is done by using the view's share method. This allows you to make a variable available to all views, regardless of where they are rendered. You can share a variable by adding it to the view's share method, for example:

```
View::share('key', 'value');
```

You can then access the variable in any view using blade syntax, for example:

```
<h1>{{ $key }}</h1>
```

We should be careful about what you share with views. Avoid sharing sensitive data, such as passwords or API keys, with views as it could be visible to the end user."

Ktunotes.in