ASSIGNMENT 2: Develop Back-End using PHP

Reg No: 21RP03296

Class: Level6 Y2 A

Date: 26/01/2023

Q1)

PHP is an acronym for "PHP: Hypertext Preprocessor". It is a widely-used, open source scripting language which is executed on the server and is free to download and use.

Php files are saved with .php extention and are saved in the local server's folder eg htdocs when using Xampp. In addition, a php program is started with "<?php" and ends with "?>" .

<u>Syntax:</u>

```
<?php

Block of codes to be executed

?>
```

Q2)

There are a number of reasons to why we need to use php programming language including the following:-

- PHP runs on various platforms (Windows, Linux, Unix, Mac OS X, etc.)
- PHP is compatible with almost all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is free. Hence affordable.
- PHP is easy to learn and runs efficiently on the server side

Q3)

The latest PHP version is PHP8.2 and its features are:-

- Readonly classes

- New memory_reset_peak_usage Function
- Null and False Standalone Types
- Deprecate Partially Supported Callables
- New /n Modifier
- Deprecate ${} String Interpolation
- Other Minor Improvements
- Random Extension Improvement
- Additional RFCs in PHP 8.2

Q4)

NEW RELEASE	**STABLE RELEASE**
Is the distribution of the final version or the newest version of a software product	Is a version that has been tested as thoroughly as possible and is as reliable as we can make it.
The software product may be public or private and generally signifies the unveiling of a new or upgraded version of the product	It does not have all the features of a beta release and it does not have the latest fixes for problems

Q5)

The main features of PHP are

- PHP can generate dynamic page content.
- PHP can create, open, read, write, delete, and close files on the server.
- PHP can collect form data.
- PHP can send and receive cookies.
- PHP can add, delete, modify data in your database.

- PHP can be used to control user-access.
- PHP can encrypt data.

Q6)

In PHP, variables and constants are case sensitive, while functions are not case sensitive.

Example: $a and $A are two different variable names

ie;

```
// 👇 you can create two variables like this:
$num = 99;
$NUM = 20;

echo $num; // 99
echo "\n".$NUM; // 20
```

You cannot display the value of $num by writing “echo $NUM” because they are different.

Q7)

We use comments to:

- ✓ Let others understand your code
- ✓ Remind yourself of what you did

We have two types of comments namely single-line and multiple-line comments

Single-line comments: They are used to comment a single or one line of code

Ex:

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

<?php
// This is a single-line comment

# This is also a single-line comment
?>

</body>
</html>
```

Multiple-line comments: They are used to comment more than one line of code or a block of codes.

Ex:

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

<?php
/*
This is a multiple-lines comment block
that spans over multiple
lines
*/
?>

</body>
</html>
```

Q8)

a. Echo() vs print(): Echo and print are more or less the same. They are both used to output data to the screen. The differences are small: echo has no return value while print has a return value of 1 so it can be used in expressions. echo can take multiple parameters (although such usage is rare) while print can take one argument.

b. Print() vs printf(): The two PRINT procedures perform formatted output. PRINT performs output to the standard output stream , while PRINTF requires a file unit to be explicitly specified.

c. Printf() vs print_r(): The printf( ) function builds a formatted string by inserting values into a template. The print_r( ) function is useful for debugging—it prints the contents of arrays, objects, and other things, in a more-or-less human-readable form.

d. Print_r() vs var_dump(): The var_dump() function displays structured information about variables/expressions including its type and value. Whereas The print_r() displays information about a variable in a way that's readable by humans.

Q9)

- ✓ **Scalar datatypes**

A scalar data type is something that has a finite set of possible values, following some scale, i.e. each value can be compared to any other value as either equal, greater or less..

- **boolean.**
- **integer.**
- **float.**
- **string.**

✓ **Compound datatypes**

A composite or compound data type is built by combining primitive data types.

- Lists
- Hashes
- Structures. ...
- Pointers. ...
- Object References**.**

✓ **Special datatypes**

Special data types are specified by the programming language.

- Nominal
- Ordinal
- Discrete
- Continuous

Q10)

Php variable: is container for storing information.

<u>Variable naming conventions/rules</u>

- variable starts with the `$` sign, followed by the name of the variable
- A variable name must start with a letter or the underscore character
- A variable name cannot start with a number
- A variable name can only contain alpha-numeric characters and underscores (A-z, 0-9, and _)
- Variable names are case-sensitive (`$age` and `$AGE` are two different variables)

Q11)

Super global variables: They are variables 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.

1. **$GLOBALS**

   $GLOBALS is a PHP variable that is used in accessing other global variables within a PHP script. All the PHP global variables are kept in an array known as `$GLOBALS[index]`. The `index` holds the `variable name`.

2. **$_SERVER**

   `$_SERVER` is a superglobal that keeps information headers, paths, and locations of a PHP script.

3. **$_GET**

   The `$_GET` variable is a PHP superglobal that collects data from an HTML form after submission. The HTML form is structured in a way that `$_GET` is used as a method. `$_GET` can also be used for retrieving data sent in a `uniform resource locator`.

4. **$_POST**

   `$_POST` collects values from a HTML form. Information sent using this method is not displayed in the URL. There is also no limit to the number of characters that can be sent at a time.

5. **$_REQUEST**

   The `$_REQUEST` variable is a PHP superglobal that is used to collect data after submitting a form. It contains the contents of `$_GET`, `$_POST` and even `$_COOKIE` by default. Data from various fields can be collected by PHP using the `$_REQUEST` variable.

6. **$_SESSION**

A `$_SESSION` variable is a PHP superglobal that stores and avails information about a site user every time the user opens the site until its closure.

7. **$_COOKIE**

A cookie is a small file that is stored in a user's computer by the server. It identifies the user. Whenever a request is made to a server. A cookie is usually sent alongside the request. PHP creates cookies using the `setcookie()` function.

8. **$_FILE**

`$_FILES` is a variable that contains items that are uploaded using HTTP POST method.
The `$_FILES` array contains several elements which are stated below:

- $_FILES['file']['name'] - This is usually the original name of file to be uploaded.
- $_FILES['file']['type'] - This refers to the type of the file being uploaded.
- $_FILES['file']['size'] - The file size in bytes.
- $_FILES['file']['tmp_name'] - It refers to a temporary filename of the storage file uploaded on the server.
- $_FILE['file']['error']- The file upload's associated error code.

9. **$_ENV**

It stores environment variables available to current script. $HTTP_ENV_VARS also contains the same information, but is not a super global, and now been deprecated. Environment variables are imported into global namespace

REFERENCES

https://www.w3schools.com/

https://www.cloudways.com/blog/php-8-2

https://sebhastian.com/is-php-case-sensitive

https://www.section.io/engineering-education/php-superglobals/