Names: MANISHIMWE Patrick

Regnumber: 21RP03787

Class: IT-B

Years of study: Level 6 year 2

Module name: Develop Back-end using PHP

Module code: ITLBP601

Date: 26 January 2023

ASSIGNMET2

Q1.

PHP stands for Hypertext Preprocessor. PHP is an open-source, interpreted, and object-oriented scripting language that can be executed at the server-side.

PHP is well suited for web development. Therefore, it is used to develop web applications (an application that executes on the server and generates the dynamic page.

- PHP is an interpreted language, i.e., there is no need for compilation.
- PHP is faster than other scripting languages, for example, ASP and JSP.
- PHP is a server-side scripting language, which is used to manage the dynamic content of the website.
- PHP can be embedded into HTML.
- PHP is an object-oriented language.
- PHP is an open-source scripting language.
- PHP is simple and easy to learn language.

Q2.

The reason why we need use PHP programming it's because of , PHP is a server-side scripting language, which is used to design the dynamic web applications with MySQL database .

- It help to handles dynamic content, database as well as session tracking for the website.
- With PHP You can create sessions.
- With PHP you can access cookies variable and also set cookies.
- It helps to encrypt the data and apply validation.
- PHP supports several protocols such as HTTP, POP3, SNMP, LDAP, IMAP, and many more.
- Using PHP language, you can control the user to access some pages of your website.
- As PHP is easy to install and set up, this is the main reason why PHP is the best language to learn.
- PHP can handle the forms, such as - collect the data from users using forms save it into the database, and return useful information to the user. For example - Registration form.

Q3.

PHP 8.2 is the latest PHP version, which brings readonly **classes**, **DNF types**, **null**, **false**, and **true types**, **sensitive parameter redaction support**, a **new random extension**, and several new features along with a few deprecations.

PHP 8.1, released in 2021, brings major new features such as **Enums, Fibers, never return type, Intersection Types, readonly properties**, and more, while ironing out some of its undesired legacy features by deprecating them.

PHP 8.0, on the 25th year of PHP history, brings several important features such as **Union Types, JIT, Constructor Property Promotion, Match Syntax, Named Parameters, and several more performance, syntax, and quality-of-life** improvements.

Q4.

A **new release** is the distribution of the final version or the newest version of a software application. While A **stable release** is a version that has been tested as thoroughly as possible and is as reliable as we can make it.

Stable release it does not have all the new features of a beta release and it does not have the latest fixes for problems. While new release it have all the new features of a beta release and it have the latest fixes for problems.

Q5.

PHP is very popular language because of its simplicity and open source. There are some important features of PHP given below:

- Open Source
- Performance
- Familiarity with syntax
- Embedded
- Platform Independent
- Database Support
- Error Reporting
- Loosely Typed Language
- Web servers Support
- Security
- Control
- A Helpful PHP Community

Q6.

As you know in PHP you declare variable that are represented by a dollar sign followed by the name of the variable. PHP constructs, function names, class names are case-insensitive, whereas variables are case-sensitive.

So by example: if you write variable name like **$name** and **$NAME** or **$STUDENT** and **$student** can have different values. However, when you declare two functions with the same name, PHP produces a fatal error: cannot redeclare the function. So that why PHP is, case sensitive.

Q7.

A comment in PHP code is a line that is not executed as a part of the program. Its only purpose is to be read by someone who is looking at the code.

Comments can be used to:

Let others understand your code.
Remind yourself of what you did.

In PHP there are two types of comments such as:

1. PHP Single Line Comments

There are two ways to use single line comments in PHP.

Example:

```
<? Php

// this is C++ style single line comment
# this is Unix Shell style single line comment
echo "Welcome to PHP single line comments";

?>
```

2. PHP Multi Line Comments

In PHP, we can comments multiple lines also. To do so, we need to enclose all lines within /* */.

Let's see a simple example of PHP multiple line comment.

```
<?php

/*
Anything placed
Within comment
Will not be displayed
On the browser;
```

```
*/
echo "Welcome to PHP multi line comment";

?>
```

Q8. The defferent between php out put function :

a. Echo () vs print () : 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. **echo** is marginally faster than **print.**

```
Examples of echo : <?php
echo "<h2>My First PHP Code</h2>";
?>
Output: My First PHP Code
```

```
Examples of print : <?php
print "<h2>My First PHP Code</h2>";
?>
Output: My First PHP Code
```

b. Print() vs printf():They are all used to print text on the screen but printf function is way more complicated. I believe print is simply for text, While printf is for Formatting text with some HTML markup (F = Formatting).

```
Examples of print : <?php
print "<h2>My First PHP Code</h2>";
```

```
?>
```

Output: **My First PHP Code**

Example of printf function in PHP

```
<? Php
$number = 123;
printf("With 2 decimals: %1\$.2f
<br>With no decimals: %1\$u",$number);
?>
```

```
Output:  With 2 decimals: 123.00
         With no decimals: 123
```

c. Printf() vs print_r(): printf is for Formatting text with some HTML markup (F = Formatting). While print_r() is used to Print the information about some variables in a more human-readable way.

Example of print_r() function in php:

```
<?php
$a = array("red", "green", "blue");
print_r($a);

echo "<br>";

$b = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");
print_r($b);
?>
```

Output: Array ([0] => red [1] => green [2] => blue)

Array ([Peter] => 35 [Ben] => 37 [Joe] => 43) and for printf() function you can you the given example used above.

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.

Example of var_dump () function in php:

```
<? php
$a = 32;
echo var_dump($a) . "<br>";
?>
```

Output: int(32)

Q9. PHP Data Types:

a. **Scalar Types**

In simple words, a variable is called scalar type if it holds singular value only.

There are 4 scalar data types in PHP:

- **Boolean**: is a variable that can have one of two possible values, true or false. Boolean is the simplest data type in PHP. It holds either true or false. To specify a boolean, you can use constants true and false (Both are case-insensitive

- **Integer: is** a number without any decimal part. An integer data type is a non-decimal number between -2147483648 and 2147483647 in 32 bit systems, and between -9223372036854775808 and 9223372036854775807 in 64 bit systems.

- **Float: is** used to set fractional values. A float is a number with a decimal point and can be extended to exponential form. Float is also called a floating-point number. Various ways to represent float values are 3.14, 4.75, 5.88E+20, etc

- **String: is** series of characters, where a character is the same as a byte. This means that PHP only supports a 256-character set, and hence does not offer native Unicode support.

b. **Compound Types**

In contrast to Scalar data types, a variable is called compound if it holds multiples values within.

There are 2 compound data types in PHP.

- **Array: is** a special variable that we use to store or hold more than one value in a single variable without having to create more variables to store those values. To create an array in PHP, we use the array function array( ) . By default, an array of any variable starts with the 0 index.

- **Object:** is a compound data type (along with arrays). Values of more than one types can be stored together in a single variable. Object is an instance of either a built-in or user defined class.

c. Special Types

There are 2 special data types in PHP.

- **Resource: is** a special data type that refers to any external resource. A resource variable acts as a reference to external source of data such as stream, file, database etc.

- **NULL: is**_null () function checks whether a variable is NULL or not. This function returns true (1) if the variable is NULL, otherwise it returns false/nothing.

Q10.

A. php variable:
are characters that stores value or information such as text or integers in your code.

B. Rules for PHP variables:

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

The PHP superglobal variables are:

- $GLOBALS
- $_SERVER
- $_REQUEST
- $_POST
- $_GET
- $_FILES
- $_ENV
- $_COOKIE
- $_SESSION

1. **$GLOBALS** is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods) .PHP stores all global variables in an array called $GLOBALS[index].
2. **$_SERVER** is a PHP super global variable, which holds information about headers, paths, and script locations.
3. **$_REQUEST** is a PHP super global variable, which is used to collect data after submitting an HTML form.
4. **$_POST** is a PHP super global variable which is used to collect form data after submitting an HTML form with method="post". $_POST is also widely used to pass variables.
5. **$_GET** is a PHP super global variable which is used to collect form data after submitting an HTML form with method="get". $_GET can also collect data sent in the URL.

6. **$_FILES** is an associative array containing items uploaded via HTTP POST method. Uploading a file requires HTTP.
7. **$_ENV** is another superglobal associative array in PHP. It stores environment variables available to current script.
8. **$_COOKIE** stores variables passed to current script along with HTTP request in the form of cookies.
9. **$_SESSION** is an associative array that contains all session variables. It is used to set and get session variable values. Example: Store information.

REFERENCES

https://www.w3schools.com/php/php_superglobals_request.asp
https://www.javatpoint.com/php-tutorial
https://www.w3schools.com/php/php_intro.asp
https://www.geeksforgeeks.org/php-unique-features/
https://php.watch/