Module name: Develop back-end using PHP

Module code: ITLBP601

Regno: 21RP03471

Date: 26/01/2023

Assignment

1. What is PHP

PHP is an open source server-side scripting language that is powerful tool for making dynamic and interactive Web pages
PHP is a widely-used, free, and efficient alternative to competitors such as Microsoft's ASP. It is deep enough to run large social networks and also easy enough to be a beginner's first server side language.

PHP can generate dynamic page content, (create, open, read, write, delete, and close) files on the server, collect form data, send and receive cookies. PHP also can add, delete, modify data in your database. PHP can be used to control user-access and can encrypt data. It execute codes through server and return result as plain html on browser.

PHP file extension is (.php).

Structute of php is as flow:

```
<?php

//codes to be executed

?>
```

2. key points why do we need PHP

- Easy and Simple to Learn. PHP is considered one of the easiest scripting languages. ...

- Extremely Flexible.
- Easy Integration and Compatibility.
- Efficient Performance.
- Cost-Efficient.
- Gives Web Developer More Control……

3. The latest version of PHP

the latest version of PHP is PHP 8.2 which is released on December 8, 2022.

4. New release vs stable release

A new Release means the latest version of the Software offered for general commercial distribution at a given point in time. while **A stable release** is a version of a software package that has been tested and verified. It is the latest (and sometimes final version) of a program that is considered safe for public use.

5. PHP advanced features

1. PECL Extension

Several instruments require the native extension to be included, including timing of Redis, Elasticsearch, and Memcached.

2. Deploy Tracking

Scout can track deploys, making it easier to correlate specific deploys to changes in performance.

3. Request Queuing

Our PHP integration can measure the time it takes a request to reach your application from farther upstream (a load balancer or web server). This appears in Scout as “Request Queueing” and provides an indication of your application’s capacity. Large request queueing time is an indication that your app needs more capacity.

4. Custom Context

Context lets you see the key attributes of requests. For example, you can add custom context to answer critical questions like:

Which plan was the customer who had a slow request on?

How many users are impacted by slow requests?

How many trial customers are impacted by slow requests?

How much of an impact are slow requests having on our highest paying customers?

6. In PHP, class names as well as function/method names are case-insensitive, but it is considered good practice to them functions as they appear in their declaration. In the following example, the function that is defined as exampleFunction() is called as ExampleFunction(), which is correct because in function creation, function name start with small letter and in function call, name of function start with capital letter.

   ex :

```
        <?php
function exampleFunction($a) {
    echo $a;
}


ExampleFunction('Foo');


        ?>
```

7. 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.

   ex: <?php

```
// This is a single-line comment
# This is also a single-line comment
/*
This is a multiple-lines comment block
that spans over multiple
lines
*/
?>
```

types of comments in php

- single line comment

  The one-line comment is placed at the end of the line or at the current block.

  A one-line comment starts with the pound (#) or double forward-slash (//). The rest of the text after the (//) is ignored by the PHP interpreter.

  ex: **<?php**

  ```

  $rate = 100;
  $hours = 173;
  $payout = $hours * $rate; // payout calculation
  ?>
  ```

- multi-lines comment

  The multi-line comment is placed at the end of the more than one line or at the end of block lines.

  A Multi-line comment start with /* and end with */. For example:

  ```
  <?php
  ```

```
/*
    This is an example of a multi-line comment,
    which can span multiple lines.
*/
```

8. a) echo vs print

- 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

b) print vs printf

- printf() outputs a formatted string whereas print() outputs one or more strings.

```
ex:     <?php
        print "Hello, world!";
        ?>
        output:         Hello, world!
ex:     <?php
        $number = 8;
            $str = "Solar System";
            printf("There are %u planets in the %s.", $number, $str);
            ?>
            output:     There are 8 planets in the Solar System.
```

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.

ex1:

```
<?php

$number = 8;

$str = "Solar System";

printf("There are %u planets in the %s.", $number, $str);

?>
```

output: There are 8 planets in the Solar System.

ex2:

```
class P {

   var $name = 'nat';

   // ...

}


$p = new P;

print_r($p);
```

output

```
Object


(


    [name] => nat
)
```

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.

$arr = array ('xyz', false, true, 99, array('50'));

output for **print_r($arr)**

```
Array

(

    [0] => xyz

    [1] =>

    [2] => 1

    [3] => 99

    [4] => Array

        (

            [0] => 50

        )

)
```

output for **var_dump($arr)**

```
array(5) {

  [0]=>

  string(3) "xyz"

  [1]=>

  bool(false)
  [2]=>
  bool(true)
  [3]=>
  int(100)
  [4]=>
  array(1) {
    [0]=>
    string(2) "50"
  }
}
```

9.

- scalar data types in php
  1. boolean
  2. integer
  3. float
  4. string
- coumpound data
  5. Array.
  6. Object.
- special

  String type

  The string data type is a representation of string literals. Unlike in C, the string is a basic data type in Vue

  List type

  A variable of type list collects a set of integral type values. The list type is an abstract data type and you cannot use a list variable directly with the standard C unary or binary operators.

Associative array type

An associative array is a map or look-up table consisting of a collection of keys and their associated values. There is a one-to-one mapping between a set of keys and values. Associative arrays are supported by Perl, ksh93, and several other languages.

Timestamp data type

A variable of type probev_timestamp_t holds the return value from the timestamp ProbeVue function, which returns a timestamp in internal AIX® format.

File path data type

A variable of type path_t can be used to hold the value of __file->path (refer to __file built in for I/O probe manager) or function fd_path(). Only local or global variables of type path_t are supported. A variable of this type can also be the key or value in an associative Array.

MAC address data type

A variable of type mac_addr_t is used to hold the value of MAC address. The MAC address data type is an abstract data type and you cannot use it directly with standard C unary or binary operators.

IP address data type

This is an abstract data type and cannot be used directly with standard C unary or binary operators. Only local or global variables of type ip_addr_t are supported. A variable of this type can also be stored in an associate array either as a key or as a value.

net_info_t data type

The net_info_t variable is structure or composite variable that is used to hold the network four tuples (local and remote IP addresses and port numbers) information from the specific socket descriptor through the sockfd_netinfo Vue function.

10. Variables and variable naming rules in PHP

PHP variables are characters called containers that stores value or information such as text or integers in your code. It is important to know that variables in PHP are usually represented by a dollar sign($) followed by the name of the variable.

Variable naming rules:

- A variable name must only contain alphabets, digits, and underscore.
- Variable in PHP must start with underscore followed by variable name
- A variable name must start with an alphabet or an underscore only. It cannot start with a digit.
- No whitespace is allowed within the variable name.
- A variable name must not be any reserved word or keyword.

11. PHP super-globals

i. **$GLOBALS** is the superglobal variable that stores all user-defined global variables. The global variable names act as keys to their values.

ii. **$_SERVER** contains data about headers, scripts, and paths. The keys to the values in this array are predefined.

iii. **$_REQUEST** stores data input in the form of HTTP POST, GET and Cookies. The keys to this array are defined in the HTTP requests.

iv. **$_POST** stores data input in the form of POST requests. The keys to this array are defined in the HTTP POST request.

v. **$_GET** has data input in the form of GET requests. The keys to this array are defined in the HTTP GET request.

vi. **$_FILES** is a two-dimensional associative array that contains a list of files that were uploaded to the script using the POST method. The keys to this array are the names of the fields uploading the files and the data being accessed. For example, $_FILES[fileUploaded][name] accesses the name of the file being uploaded from the *fileUploaded* field.

vii. **$_COOKIES** keeps data input via HTTP Cookies. The keys to this array are defined when the cookies are set.

viii. **$_SESSION** holds session variables. Session variables can be accessed on multiple pages. This array's keys are defined by the users when they define session variables.

ix. **$_ENV** contains information about the environment that PHP is running in. The keys to the values in this array are predefined.

End

References

https://www.w3schools.com/php/php_intro.asp

https://www.goodworklabs.com/top-6-advantages-of-php-over-other-programming-languages/

https://stitcher.io/blog/new-in-php-82#:~:text=PHP%208.2%20is%20released%20on,and%20deprecations%20one%20by%20one.

https://www.lawinsider.com/dictionary/current-release#:~:text=Current%20Release%20means%20the%20latest,time%2C%20including%20all%20New%20Releases.

https://theproductmanager.com/topics/software-release-life-cycle/#:~:text=A%20stable%20release%20is%20a,called%20a%20%E2%80%9CStable%E2%80%9D%20release.

https://www.jetbrains.com/help/phpstorm/php-case-mismatch-in-method-call-or-class-usage.html

https://www.w3schools.com/php/php_comments.asp#:~:text=A%20comment%20in%20PHP%20code,Let%20others%20understand%20your%20code

https://www.phptutorial.net/php-tutorial/php-comments/

https://www.w3schools.com/php/php_echo_print.asp#:~:text=echo%20and%20print%20are%20more,print%20can%20take%20one%20argument

https://interviewbazaar.net/interview-question/what-is-the-difference-between-printf-and-print-in-php/#:~:text=May%204%2C%202021,outputs%20one%20or%20more%20strings.

https://www.oreilly.com/library/view/programming-php/1565926102/ch04s02.html#:~:text=The

%20printf(%20)%20function%20builds%20a,%2Dless%20human%2Dreadable%20form

https://stackoverflow.com/questions/3406171/php-var-dump-vs-print-r#:~:text=The%20var_dump()%20function%20displays,way%20that's%20readable%20by%20humans