DAILY ASSESSMENT FORMAT

Date:	12 June 2020	Name:	Karegowda kn
Course:	MySql	USN:	4al16ec029
Topic:	PHP sequrity,creating a custom error handler,types of error,PHP functions	Semester & Section:	6 sem B sec
Github Repository:	Karegowda-courses		

AFTERNOON SESSION DETAILS

Report--

PHP sequrity:

With PHP security, there are two sides to error reporting. One is beneficial to increasing security, the other is detrimental. Regardless of the method of error handling, the ability to probe a system for errors leads to providing an attacker with more information.

PHP Error Handling:

When creating scripts and web applications, error handling is an important part. If your code lacks error checking code, your program may look very unprofessional and you may be open to security risks.

Some of the most common error checking methods in PHP.

We will show different error handling methods:

- Simple "die()" statements
- Custom errors and error triggers
- Error reporting

Example:

```
<?php
if(file_exists("mytestfile.txt")) {
  $file = fopen("mytestfile.txt", "r");
} else {
  die("Error: The file does not exist.");
}
?>
```

Creating a Custom Error Handler:

Creating a custom error handler is quite simple. We simply create a special function that can be called when an error occurs in PHP.This function must be able to handle a minimum of two parameters (error level and error message) but can accept up to five parameters (optionally: file, line-number, and the error context):

Syntax:

```
error_function(error_level,error_message,
error_file,error_line,error_context)
```

Types of error

Basically there are four types of errors in PHP, which are as follows:

- Parse Error (Syntax Error)
- Fatal Error
- Warning Error
- Notice Error

1. Parse Errors (syntax errors)

The parse error occurs if there is a syntax mistake in the script; the output is Parse errors. A parse error stops the execution of the script. There are many reasons for the occurrence of parse errors in PHP. The common reasons for parse errors are as follows:

Common reason of syntax errors are:

- Unclosed quotes
- Missing or Extra parentheses
- Unclosed braces
- Missing semicolon

Example

```
<?php
echo "Cat";
echo "Dog"
echo "Lion";
?>
```

2. Fatal Errors

Fatal errors are caused when PHP understands what you've written, however what you're asking it to do can't be done. Fatal errors stop the execution of the script. If you are trying to

access the undefined functions, then the output is a fatal error.

Example

```
<?php
function fun1()
{
echo "Vineet Saini";
}
fun2();
echo "Fatal Error !!";
?>
```

3.Warning Errors

Warning errors will not stop execution of the script. The main reason for warning errors are to include a missing file or using the incorrect number of parameters in a function.

Example

```
<?php
echo "Warning Error!!";
include ("Welcome.php");
?>
```

4. Notice Errors

Notice that an error is the same as a warning error i.e. in the notice error execution of the script does not stop. Notice that the error occurs when you try to access the undefined variable, then produce a notice error.

Example

```
<?php
$a="Vineet kumar saini";
echo "Notice Error !!";
echo $b;
?>
```

PHP Built-in Functions:

PHP has over 1000 built-in functions that can be called directly, from within a script, to perform a specific task.

PHP User Defined Functions:

Besides the built-in PHP functions, it is possible to create your own functions.

- A function is a block of statements that can be used repeatedly in a program.
- A function will not execute automatically when a page loads.
- A function will be executed by a call to the function.

Create a User Defined Function in PHP:

A user-defined function declaration starts with the word function:

Syntax

```
function functionName() {
  code to be executed;
}
```