Web Design and Web Development

Assignment 3 - Unit 3

71. Tell the different file handling modes available in PHP?

```
<?php
echo "=== Demonstrating PHP File Handling Modes (Simulated) ===\n\n";
// 1. Write mode (w) - creates or truncates
$file = fopen("php://memory", "w+");
fwrite($file, "This is written in w mode.\n");
rewind($file);
echo "w mode output:\n" . stream_get_contents($file) . "\n\n";
// 2. Append mode (a) - adds to end
$file = fopen("php://memory", "a+");
fwrite($file, "First line in a mode.\n");
fwrite($file, "Appended line in a mode.\n");
rewind($file);
echo "a mode output:\n" . stream_get_contents($file) . "\n\n";
// 3. Read mode (r) - must exist (simulate by writing first)
$temp = fopen("php://memory", "w+");
fwrite($temp, "Reading in r mode.\n");
rewind($temp);
echo "r mode output:\n" . fread($temp, 1024) . "\n\n";
// 4. Read & Write mode (r+)
$temp = fopen("php://memory", "w+");
fwrite($temp, "Original text.\n");
rewind($temp);
fwrite($temp, "Edited in r+ mode.\n"); // Overwrites from start
rewind($temp);
```

```
echo "r+ mode output:\n" . stream_get_contents($temp) . "\n\n";
// 5. Exclusive create mode (x) - simulated
echo "x mode simulation: Creates a new file only if it does not exist.\n";
echo "(In online compilers, actual file creation is restricted.)\n";
?>
```

OUTPUT:

```
w mode output:
This is written in w mode.
a mode output:
First line in a mode.
Appended line in a mode.
r mode output:
Reading in r mode.
r+ mode output:
Edited in r+ mode.
x mode simulation: Creates a new file only if it does not exist.
(In online compilers, actual file creation is restricted.)
```

72. Write the PHP program to implement the following session functions.

(i)session_register() (ii) session_unset() (iii) session_destroy()

```
<?php
session_start(); // Start the session

// (i) session_register() equivalent
// Old: session_register("username");
// New way:
$_SESSION["username"] = "JohnDoe";
$_SESSION["email"] = "john@example.com";
```

```
echo "<h3>Session Variables Registered:</h3>";

print_r($_SESSION);

// (ii) session_unset() - remove all session variables but keep session

session_unset();

echo "<br><br><strong>After session_unset():</strong><br>";

print_r($_SESSION);

// (iii) session_destroy() - remove session data completely

session_destroy();

echo "<br><br>Session destroyed.";

?>
```

OUTPUT:

```
<h3>Session Variables Registered:</h3>Array

(

    [username] => JohnDoe

    [email] => john@example.com

)

<br><br><strong>After session_unset():</strong><br>Array

(

)

<br><br>Session destroyed.
```

73. Develop a PHP program to display an error message when a file other than jpg or png is uploaded.

```
<?php

// Simulated uploaded file name

$uploadedFileName = "sample.gif"; // Change this to test: e.g., "photo.jpg", "image.png"
```

```
// Allowed extensions
$allowedExtensions = ['jpg', 'jpeg', 'png'];

// Extract file extension (lowercase)
$fileExtension = strtolower(pathinfo($uploadedFileName, PATHINFO_EXTENSION));

// Check file type
if (in_array($fileExtension, $allowedExtensions)) {
   echo "✅ File accepted: $uploadedFileName\n";
} else {
   echo "❌ Error: Only JPG or PNG files are allowed! ($uploadedFileName)\n";
}
?>
```

OUTPUT:

```
ERROR!
❌ Error: Only JPG or PNG files are allowed! (sample.gif)
```

74. Outline the file reading and writing functions of PHP with suitable examples.

```
<?php
// Open an in-memory file for read/write
$fp = fopen("php://memory", "w+");

// =======================
// 1. WRITE TO FILE (fwrite)
// =======================
fwrite($fp, "Hello, this is PHP file handling in memory!\n");
fwrite($fp, "We are writing this text into a memory file.\n");
```

```
// Move pointer back to start for reading
rewind($fp);
// =======================
// 2. READ FILE LINE BY LINE (fgets())
// =======================
echo "Reading file line-by-line using fgets():\n";
while (!feof($fp)) {
  echo fgets($fp);
}
// =======================
// 3. APPEND TO FILE
// =======================
fseek($fp, 0, SEEK_END); // Move to end
fwrite($fp, "This line is appended at the end.\n");
// Move pointer back to start for final reading
rewind($fp);
echo "\nReading whole file after appending:\n";
while (!feof($fp)) {
  echo fgets($fp);
}
fclose($fp);
?>
```

OUTPUT:

```
Reading file line-by-line using fgets():
Hello, this is PHP file handling in memory!
We are writing this text into a memory file.


Reading whole file after appending:
```

```
Hello, this is PHP file handling in memory!

We are writing this text into a memory file.

This line is appended at the end.

=== Code Execution Successful ===
```

75. Build a PHP function that takes the filename and a new line of content as input. Append the new line of content as a new line at the end of the file.

```php
<?php

// Function to append a line to a "file" (in memory)

function appendLineToFile($filename, $newLine) {

  // Open memory-based file

  $fp = fopen($filename, "a+"); // a+ = read/write and append mode


  if (!$fp) {

    echo "Unable to open file!";

    return;

  }

  // Append new line

  fwrite($fp, $newLine . "\n");

// Move pointer to start to read file contents

  rewind($fp);


  // Display updated file content

  echo "File content after appending:\n";

  while (!feof($fp)) {

    echo fgets($fp);

  }

  fclose($fp);
```

```
// --------------------
// Test the function
// --------------------
// Use php://memory instead of demo.txt (safe for online compilers)
$filename = "php://memory";

// First, create the file with some content
$fp = fopen($filename, "w+");
fwrite($fp, "First line in file.\n");
fclose($fp);

// Append new line using the function
appendLineToFile($filename, "This is the appended line");
?>
```

OUTPUT:

```
File content after appending:
This is the appended line
```

76. Write a simple 'birthday countdown' script, the script will count the number of days between current day and birthday.

```
<?php
// Set your birthday (month and day)
$birthdayMonth = 12;
$birthdayDay = 25;

// Get today's date
$today = new DateTime();

// Create birthday date for this year
```

```
$birthdayThisYear = new DateTime($today->format('Y') . "-$birthdayMonth-$birthdayDay");

// If birthday already passed this year, use next year
if ($birthdayThisYear < $today) {
   $birthdayNext = new DateTime(($today->format('Y') + 1) . "-$birthdayMonth-$birthdayDay");
} else {
   $birthdayNext = $birthdayThisYear;
}

// Calculate difference
$interval = $today->diff($birthdayNext);
$daysLeft = $interval->days;

echo "Your birthday is in $daysLeft day(s)!";
?>
```

OUTPUT:

```
Your birthday is in 131 day(s)!
```

77. Classify the various network functions which provide network information or functions.

```
<?php
// Define network functions
$networkFunctions = [
   "Addressing & Name Resolution" => [
      "Examples" => ["DNS", "DHCP", "ARP"],
      "Purpose" => "Identify devices and map names to addresses."
```

```
    ],
    "Routing & Forwarding" => [
        "Examples" => ["RIP", "OSPF", "BGP"],
        "Purpose" => "Determine and forward data packets through the best path."
    ],
    "Monitoring & Management" => [
        "Examples" => ["SNMP", "NetFlow", "Ping", "Traceroute"],
        "Purpose" => "Monitor network devices, traffic, and performance."
    ],
    "Security" => [
        "Examples" => ["Firewall", "IDS/IPS", "VPN"],
        "Purpose" => "Protect network from unauthorized access and attacks."
    ],
    "Quality of Service (QoS)" => [
        "Examples" => ["Traffic shaping", "Latency monitoring", "Jitter control"],
        "Purpose" => "Prioritize and manage network traffic for critical applications."
    ],
    "Discovery & Topology" => [
        "Examples" => ["LLDP", "CDP", "Nmap"],
        "Purpose" => "Discover network devices and visualize topology."
    ],
    "Service & Access" => [
        "Examples" => ["HTTP", "FTP", "Proxy servers", "Load balancers"],
        "Purpose" => "Provide network services and application access."
    ]
];

// Display network functions
```

```
foreach ($networkFunctions as $type => $details) {
   echo "=== $type ===\n";
   echo "Examples: " . implode(", ", $details["Examples"]) . "\n";
   echo "Purpose: " . $details["Purpose"] . "\n\n";
}
?>
```

OUTPUT:

```
=== Addressing & Name Resolution ===
Examples: DNS, DHCP, ARP
Purpose: Identify devices and map names to addresses.
=== Routing & Forwarding ===
Examples: RIP, OSPF, BGP
Purpose: Determine and forward data packets through the best path.
=== Monitoring & Management ===
Examples: SNMP, NetFlow, Ping, Traceroute
Purpose: Monitor network devices, traffic, and performance.
=== Security ===
Examples: Firewall, IDS/IPS, VPN
Purpose: Protect network from unauthorized access and attacks.
=== Quality of Service (QoS) ===
Examples: Traffic shaping, Latency monitoring, Jitter control
Purpose: Prioritize and manage network traffic for critical applications.
=== Discovery & Topology ===
Examples: LLDP, CDP, Nmap
Purpose: Discover network devices and visualize topology.
=== Service & Access ===
Examples: HTTP, FTP, Proxy servers, Load balancers
```

Purpose: Provide network services and application access.

78. Elucidate the steps to read and delete existing cookies in the browser.

```
<?php
// Simulate cookies in an array
$simulatedCookies = [
  "username" => "JohnDoe",
  "theme" => "dark"
];
echo "=== Reading Cookies ===\n";
foreach($simulatedCookies as $name => $value){
  echo "Name: $name, Value: $value\n";
}
// Simulate deleting cookies
$simulatedCookies = [];
echo "\n=== Cookies Deleted ===\n";
if(empty($simulatedCookies)){
  echo "No cookies found.\n";
}
?>
```

OUTPUT:

```
Name: username, Value: JohnDoe
Name: theme, Value: dark
=== Cookies Deleted ===
No cookies found.
```

79. Create a PHP program to append two files in another file.

```
<?php
// Simulated file contents
```

```
$file1Content = "This is content of file1.\nLine 2 of file1.";
$file2Content = "This is content of file2.\nLine 2 of file2.";
// Append the two "files" into another "file"
$combinedContent = $file1Content . "\n" . $file2Content;
// Display the result (simulating writing to combined.txt)
echo "=== Combined File Content ===\n";
echo $combinedContent;
?>
```

OUTPUT:

```
=== Combined File Content ===
This is content of file1.
Line 2 of file1.
This is content of file2.
Line 2 of file2.
```

80. Build a test cookie with the setcookie( ) function using PHP.

```
<?php
// Set a test cookie
$cookie_name = "TestCookie";
$cookie_value = "HelloWorld";

// Set cookie to expire in 1 hour (3600 seconds)
setcookie($cookie_name, $cookie_value, time() + 3600, "/");

// Check if cookie is set
if(isset($_COOKIE[$cookie_name])){
  echo "Cookie '{$cookie_name}' is already set. Value: " . $_COOKIE[$cookie_name];
} else {
  echo "Cookie '{$cookie_name}' has been created! Refresh the page to see it.";
}
?>
```

OUTPUT:

```
Cookie 'TestCookie' has been created! Refresh the page to see it.
```

81. Create Admin Login, Logout form using session variables in PHP.

```
<?php
// Simulated admin credentials
$admin_username = "admin";
$admin_password = "admin123";

// Simulated user input (change these to test)
$input_username = "admin"; // Replace with test username
$input_password = "admin123"; // Replace with test password
$action = "login"; // Use "login" or "logout"

// Simulated session variable
$session_admin = null;

if($action === "login"){
  if($input_username === $admin_username && $input_password ===
$admin_password){
      $session_admin = $input_username;
      echo "Login successful! Welcome, $session_admin.\n";
  } else {
      echo "Invalid username or password!\n";
```

```
    }
} elseif($action === "logout"){
    $session_admin = null;
    echo "You have been logged out.\n";
}

// Display dashboard if logged in
if($session_admin){
    echo "=== Dashboard ===\n";
    echo "Admin: $session_admin\n";
    echo "You are logged in.\n";
} else {
    echo "You are not logged in.\n";
}
?>
```

output:

```
Login successful! Welcome, admin.
=== Dashboard ===
Admin: admin
You are logged in.
```

82. Implement the setcookie( ) function with various arguments using the PHP program.

```
<?php
// Simulated cookies array
$simulatedCookies = [];
```

```
// Cookie 1: Basic cookie
$simulatedCookies['user'] = "JohnDoe";

// Cookie 2: Cookie with expiration
$simulatedCookies['session_id'] = [
    "value" => "abc123",
    "expires" => date("Y-m-d H:i:s", time() + 3600)
];

// Cookie 3: Cookie with path
$simulatedCookies['theme'] = [
    "value" => "dark",
    "path" => "/"
];

// Cookie 4: Cookie with domain
$simulatedCookies['lang'] = [
    "value" => "en",
    "domain" => "example.com"
];

// Cookie 5: Secure cookie
$simulatedCookies['secure_cookie'] = [
    "value" => "secure_value",
    "secure" => true
];

// Cookie 6: HttpOnly cookie
```

```
$simulatedCookies['httponly_cookie'] = [
    "value" => "secret",
    "httponly" => true
];

// Display simulated cookies
echo "=== Simulated Cookies ===\n\n";
foreach($simulatedCookies as $name => $data){
    echo "Name: $name\n";
    if(is_array($data)){
        foreach($data as $key => $value){
            echo "  $key: $value\n";
        }
    } else {
        echo "  Value: $data\n";
    }
    echo "\n";
}
?>
```

output:

```
=== Simulated Cookies ===

Name: user
  Value: JohnDoe
```

```
Name: session_id
 value: abc123
 expires: 2025-08-15 16:32:18

Name: theme
 value: dark
 path: /

Name: lang
 value: en
 domain: example.com

Name: secure_cookie
 value: secure_value
 secure: 1

Name: httponly_cookie
 value: secret
 httponly: 1
```

83. Develop a PHP script to accept email addresses and validate it. Display the domain name of the email and result of validation.

```
<?php
// Simulate user input (replace with any email to test)
$email = "example@gmail.com";

// Validate email
if(filter_var($email, FILTER_VALIDATE_EMAIL)){
  $validationResult = "Valid Email";
```

```
    // Extract domain
    $parts = explode("@", $email);
    $domain = isset($parts[1]) ? $parts[1] : "";
} else {
    $validationResult = "Invalid Email";
    $domain = "N/A";
}

// Display result
echo "Email: $email\n";
echo "Validation: $validationResult\n";
echo "Domain: $domain\n";
?>
```

output:

```
Email: example@gmail.com
Validation: Valid Email
Domain: gmail.com
```

84. Implement a PHP program that creates sessions, sets values in sessions, and removes data from the sessions.

```
<?php
// Start the session
session_start();

// Initialize messages array
$messages = [];
```

```
// 1. Set session values
if(!isset($_SESSION['user'])){
   $_SESSION['user'] = "JohnDoe";
   $messages[] = "Session variable 'user' is set to 'JohnDoe'.";
}
if(!isset($_SESSION['role'])){
   $_SESSION['role'] = "Admin";
   $messages[] = "Session variable 'role' is set to 'Admin'.";
}

// 2. Display current session data
$messages[] = "Current session data:";
foreach($_SESSION as $key => $value){
   $messages[] = "$key => $value";
}

// 3. Remove a session variable
if(isset($_SESSION['role'])){
   unset($_SESSION['role']);
   $messages[] = "Session variable 'role' has been removed.";
}

// 4. Destroy the session completely (optional, uncomment to test)
// session_unset();
// session_destroy();
// $messages[] = "All session data destroyed.";

// Display messages
```

```
echo "=== Session Demo ===\n";
foreach($messages as $msg){
   echo $msg . "\n";
}
?>
```

output:

```
Session variable 'user' is set to 'JohnDoe'.
Session variable 'role' is set to 'Admin'.
Current session data:
user => JohnDoe
role => Admin
Session variable 'role' has been removed.
```

85. List the steps to get file name from a path in PHP.

```
<?php
// Example file path
$path = "/home/user/documents/report.pdf";

// 1. Get the full file name
$filename = basename($path);

// 2. Get file name without extension
$filenameWithoutExt = pathinfo($path, PATHINFO_FILENAME);

// 3. Get file extension
$fileExtension = pathinfo($path, PATHINFO_EXTENSION);
```

```
// 4. Get directory name
$directory = pathinfo($path, PATHINFO_DIRNAME);

// Display results
echo "File Path: $path\n";
echo "Full File Name: $filename\n";
echo "File Name without Extension: $filenameWithoutExt\n";
echo "File Extension: $fileExtension\n";
echo "Directory: $directory\n";
?>
```

output:

```
File Path: /home/user/documents/report.pdf
Full File Name: report.pdf
File Name without Extension: report
File Extension: pdf
Directory: /home/user/documents
```

86. PHP File functions with example program.

```
<?php
// Set your birthday (format: YYYY-MM-DD)
$birthday = "2000-08-20";

// Get current date
$today = date("Y-m-d");

// Convert to DateTime objects
$todayObj = new DateTime($today);
```

```
$birthdayObj = new DateTime(date("Y") . "-" . date("m-d", strtotime($birthday)));

// If birthday has already passed this year, use next year
if($birthdayObj < $todayObj){
   $birthdayObj->modify("+1 year");
}

// Calculate difference
$interval = $todayObj->diff($birthdayObj);
$daysRemaining = $interval->days;

// Output
echo "Today: " . $todayObj->format("Y-m-d") . "\n";
echo "Next Birthday: " . $birthdayObj->format("Y-m-d") . "\n";
echo "Days until birthday: $daysRemaining\n";
?>
```

output:

```
Today: 2025-08-15
Next Birthday: 2025-08-20
Days until birthday: 5
```

87. Create a PHP Admin Login, Logout form using session variables.

```
<?php
// Start session
session_start();

// Hardcoded admin credentials
```

```
$admin_username = "admin";

$admin_password = "admin123";


// Messages array for output

$messages = [];


// Simulate user input (change these to test)

$input_username = "admin";  // Replace with test username

$input_password = "admin123"; // Replace with test password

$action = "login"; // "login" or "logout"


// Handle login

if($action === "login"){

  if($input_username === $admin_username && $input_password ===
$admin_password){

    $_SESSION['admin'] = $input_username;

    $messages[] = "Login successful! Welcome, $input_username.";

  } else {

    $messages[] = "Invalid username or password!";

  }

}


// Handle logout

if($action === "logout" && isset($_SESSION['admin'])){

  unset($_SESSION['admin']);

  $messages[] = "You have been logged out.";

}
```

```
// Display current session status
if(isset($_SESSION['admin'])){
   $messages[] = "Admin '{$_SESSION['admin']}' is currently logged in.";
} else {
   $messages[] = "No admin is logged in.";
}

// Output all messages
echo "=== Admin Login/Logout Simulation ===\n";
foreach($messages as $msg){
   echo $msg . "\n";
}
?>
```

output:

```
Login successful! Welcome, admin.
Admin 'admin' is currently logged in.
```

88. Generate a simple PHP 'birthday countdown' script, the script will count the number of days between current day and birthday.

```
<?php
// Set your birthday (format: YYYY-MM-DD)
$birthday = "2000-08-20";

// Get current date
$today = date("Y-m-d");

// Convert to DateTime objects
$todayObj = new DateTime($today);
```

```
$birthdayObj = new DateTime(date("Y") . "-" . date("m-d", strtotime($birthday)));

// If birthday has already passed this year, use next year
if($birthdayObj < $todayObj){
   $birthdayObj->modify("+1 year");
}

// Calculate difference
$interval = $todayObj->diff($birthdayObj);
$daysRemaining = $interval->days;

// Output
echo "Today: " . $todayObj->format("Y-m-d") . "\n";
echo "Next Birthday: " . $birthdayObj->format("Y-m-d") . "\n";
echo "Days until birthday: $daysRemaining\n";
?>
```

output:

```
Today: 2025-08-15
Next Birthday: 2025-08-20
Days until birthday: 5
```

89. Construct a PHP program to read a file using file operation.

```
<?php
// Simulated file content (instead of an actual file)
$filename = "example.txt";
$simulatedFileContent = "Hello, this is a sample file.\nThis is the second line.\n";
```

```
// Display that we are "reading" the file
echo "Reading file '$filename' (simulated):\n\n";

// Using file_get_contents simulation
$content = $simulatedFileContent; // Normally: file_get_contents($filename)
echo $content;

// Alternative: Using fopen, fread, fclose (simulated)
echo "\nReading line by line (simulated using fopen/fread):\n";

$lines = explode("\n", $simulatedFileContent); // simulate reading lines
foreach($lines as $line){
  if(!empty($line)){
    echo $line . "\n";
  }
}
?>
```

output:

```
Reading file 'example.txt' (simulated):

Hello, this is a sample file.
This is the second line.

Reading line by line (simulated using fopen/fread):
Hello, this is a sample file.
This is the second line.
```

90. Write a PHP program to open a text file and print all the data in it.

```
<?php
// Simulated text file name
$filename = "sample.txt";

// Simulated file content
$fileContent = "Hello, this is the first line of the file.\n";
$fileContent .= "This is the second line.\n";
$fileContent .= "And this is the third line.\n";

// Display message
echo "Opening file '$filename' and printing all data (simulated):\n\n";

// Print the entire file content
echo $fileContent;

// Simulate reading line by line (like fopen/fgets)
echo "\nReading line by line (simulated):\n";
$lines = explode("\n", $fileContent); // Split content into lines
foreach($lines as $line){
  if(!empty($line)){
    echo $line . "\n";
  }
}
?>
```

output:

```
Opening file 'sample.txt' and printing all data (simulated):

Hello, this is the first line of the file.
This is the second line.
And this is the third line.

Reading line by line (simulated):
Hello, this is the first line of the file.
This is the second line.
And this is the third line.
```

91. Recognize how sessions are stored in PHP?

```
<?php
// Start session
session_start();

// Set session variables
$_SESSION['username'] = 'Alice';
$_SESSION['role'] = 'Admin';

// Display session data
echo "Session Data:\n";
echo "Username: " . $_SESSION['username'] . "\n";
echo "Role: " . $_SESSION['role'] . "\n";

// To remove a session variable
unset($_SESSION['role']);
```

```
echo "\nAfter removing 'role':\n";

echo "Role: " . (isset($_SESSION['role']) ? $_SESSION['role'] : 'Not set') . "\n";


// To destroy the session completely

// session_destroy();

// echo "Session destroyed.";

?>
```

OUTPUT:

```
Session Data:

Username: Alice

Role: Admin


After removing 'role':

Role: Not set
```

92. Apply the various steps involved to get cookie value using server-side scripting.

```
<?php

// 1. Set a cookie (expires in 1 hour)

$cookie_name = "user";

$cookie_value = "Alice";

setcookie($cookie_name, $cookie_value, time() + 3600, "/"); // "/" means available across the site


// 2. Check if the cookie is available

echo "=== Cookie Example ===\n";


if(isset($_COOKIE[$cookie_name])){
  // 3. Access the cookie value

  $value = $_COOKIE[$cookie_name];

  echo "Cookie '$cookie_name' value: $value\n";

} else {

  echo "Cookie '$cookie_name' is not set yet. It will be available on next request.\n";

}

?>
```

OUTPUT:

```
Cookie 'user' is not set yet. It will be available on next request.
```

93. Write a PHP program that keeps track of how many times a visitor has loaded the page.

```
<?php

// Cookie name

$cookie_name = "visit_count";


// Check if cookie exists

if(isset($_COOKIE[$cookie_name])){

  // Increment visit count

  $visit_count = $_COOKIE[$cookie_name] + 1;

} else {

  // First visit

  $visit_count = 1;

}


// Set/update the cookie (expires in 1 year)
```

```
setcookie($cookie_name, $visit_count, time() + (365*24*60*60), "/");

// Display visit count
echo "You have visited this page $visit_count time(s).\n";
?>
```

OUTPUT:

```
You have visited this page 1 time(s).
```

94. Evaluate the different file handling modes available in PHP with appropriate examples.

```
<?php
// Simulated file content
$fileContent = "";

// 1. Simulate write mode (w) - overwrites content
$fileContent = "This is written using 'w' mode.\n";
echo "After 'w' mode:\n$fileContent\n";

// 2. Simulate append mode (a) - adds content at the end
$fileContent .= "This is appended using 'a' mode.\n";
echo "After 'a' mode:\n$fileContent\n";

// 3. Simulate read mode (r) - just display content
echo "Reading content (r mode simulated):\n$fileContent\n";

// 4. Simulate read/write mode (r+) - read and add
$fileContent .= "Adding text using 'r+' mode.\n";
```

```
echo "After 'r+' mode:\n$fileContent\n";

// 5. Simulate write/read mode (w+) - overwrites and reads
$fileContent = "Overwriting content using 'w+' mode.\n";
echo "After 'w+' mode:\n$fileContent\n";
?>
```

OUTPUT:

```
After 'w' mode:
This is written using 'w' mode.

After 'a' mode:
This is written using 'w' mode.
This is appended using 'a' mode.

Reading content (r mode simulated):
This is written using 'w' mode.
This is appended using 'a' mode.

After 'r+' mode:
This is written using 'w' mode.
This is appended using 'a' mode.
Adding text using 'r+' mode.

After 'w+' mode:
Overwriting content using 'w+' mode.
```

95. Discuss the purpose and functionality of PHP sessions and outline their potential application in maintaining a user's shopping cart state across multiple pages.

```
<?php
// Start the session
session_start();

// Initialize shopping cart if not already set
if (!isset($_SESSION['cart'])) {
   $_SESSION['cart'] = [];
}

// Simulated "page actions" - in a real site, this comes from forms/buttons
$action = $_GET['action'] ?? ''; // 'add', 'remove', or empty
$productId = $_GET['id'] ?? '';
$productName = $_GET['name'] ?? '';
$productPrice = $_GET['price'] ?? '';

// Function to add a product to the cart
function addToCart($id, $name, $price) {
   $_SESSION['cart'][] = [
      'id' => $id,
      'name' => $name,
      'price' => $price
   ];
}
```

```
// Function to remove a product by ID
function removeFromCart($id) {
  foreach ($_SESSION['cart'] as $key => $item) {
    if ($item['id'] == $id) {
      unset($_SESSION['cart'][$key]);
    }
  }
  // Reindex array
  $_SESSION['cart'] = array_values($_SESSION['cart']);
}

// Handle actions
if ($action === 'add' && $productId && $productName && $productPrice) {
  addToCart($productId, $productName, $productPrice);
}

if ($action === 'remove' && $productId) {
  removeFromCart($productId);
}

// Display shopping cart
echo "=== Your Shopping Cart ===\n";
if (empty($_SESSION['cart'])) {
  echo "Cart is empty.\n";
} else {
  $total = 0;
  foreach ($_SESSION['cart'] as $item) {
    echo $item['name'] . " - ₹" . $item['price'] . " [ID: " . $item['id'] . "]\n";
```

```
        $total += $item['price'];
    }
    echo "Total: ₹$total\n";
}

// Simulated page links (in real site, these are HTML links or buttons)
echo "\n=== Simulated Actions ===\n";
echo "Add T-shirt: ?action=add&id=101&name=T-shirt&price=299\n";
echo "Add Jeans: ?action=add&id=102&name=Jeans&price=799\n";
echo "Remove T-shirt: ?action=remove&id=101\n";
echo "Clear Cart: ?action=clear\n";

// Optional: Clear entire cart
if ($action === 'clear') {
    $_SESSION['cart'] = [];
    echo "Cart cleared.\n";
}
?>
```

OUTPUT:

```
Cart is empty.

=== Simulated Actions ===
Add T-shirt: ?action=add&id=101&name=T-shirt&price=299
Add Jeans: ?action=add&id=102&name=Jeans&price=799
Remove T-shirt: ?action=remove&id=101
Clear Cart: ?action=clear
```

96. How to register a variable in a PHP session? Give an example.

```
<?php
// 1. Start the session
session_start();

// 2. Register variables in the session
$_SESSION['username'] = "Alice";
$_SESSION['role'] = "Admin";

// 3. Access session variables
echo "Username stored in session: " . $_SESSION['username'] . "\n";
echo "Role stored in session: " . $_SESSION['role'] . "\n";
?>
```

OUTPUT:

```
Username stored in session: Alice
Role stored in session: Admin
```

97. Discover the various session functions of PHP. Explain its behavior.

```
<?php
// 1. Check session status and start session if not active
if(session_status() == PHP_SESSION_NONE){
  session_start();
  echo "Session started.\n";
}

// 2. Register variables in the session
```

```
$_SESSION['username'] = 'Alice';
$_SESSION['role'] = 'Admin';
echo "Session variables set: username = {$_SESSION['username']}, role = {$_SESSION['role']}\n";

// 3. Get current session ID and name
echo "Session ID: " . session_id() . "\n";
echo "Session Name: " . session_name() . "\n";

// 4. Regenerate session ID for security
session_regenerate_id();
echo "New Session ID after regeneration: " . session_id() . "\n";

// 5. Write and close session (saves data and unlocks)
session_write_close();
echo "Session data saved and session closed.\n";

// 6. Reopen session to access variables
session_start();
echo "Accessing session variables after reopening: username = {$_SESSION['username']}\n";

// 7. Unset session variables
session_unset();
echo "Session variables unset. Username now: " . (isset($_SESSION['username']) ? $_SESSION['username'] : 'Not set') . "\n";

// 8. Destroy the session
session_destroy();
```

```
echo "Session destroyed.\n";

?>
```

OUTPUT:

```
Session started.

Session variables set: username = Alice, role = Admin

Session ID: be8dbf956f72aff2bfa171d2cbf8c09d

Session Name: PHPSESSID


Warning: session_regenerate_id(): Session ID cannot be regenerated after headers have already been sent in /tmp/DVUGysJBtt/main.php on line 18

New Session ID after regeneration: be8dbf956f72aff2bfa171d2cbf8c09d

Session data saved and session closed.


Warning: session_start(): Session cannot be started after headers have already been sent in /tmp/DVUGysJBtt/main.php on line 26

Accessing session variables after reopening: username = Alice

Session variables unset. Username now: Alice


Warning: session_destroy(): Trying to destroy uninitialized session in /tmp/DVUGysJBtt/main.php on line 34

Session destroyed.
```

98. Model an Admin Login, Logout form using session variables in PHP.

```
<?php

// Simulate login/logout in one execution

$action = $_POST['action'] ?? '';

$username = $_POST['username'] ?? '';
```

```
$password = $_POST['password'] ?? '';

// Hardcoded credentials
$admin_user = "admin";
$admin_pass = "12345";

// Flag to show dashboard
$logged_in = false;
$message = '';

// Handle login
if ($action === "login") {
  if ($username === $admin_user && $password === $admin_pass) {
    $logged_in = true;
    $message = "✅ Login successful! Welcome, $username.<br><br>";
  } else {
    $message = "❌ Invalid username or password.<br><br>";
  }
}

// Handle logout
if ($action === "logout") {
  $message = "🔒 You have been logged out.<br><br>";
}

echo $message;

if ($logged_in) {
  echo "<h2>Admin Dashboard</h2>";
  echo "Hello, $username!<br><br>";
  echo '
  <form method="post">
    <input type="hidden" name="action" value="logout">
    <input type="submit" value="Logout">
  </form>
  ';
} else {
  echo '
  <h2>Admin Login</h2>
  <form method="post">
    <input type="hidden" name="action" value="login">
    Username: <input type="text" name="username" required><br><br>
    Password: <input type="password" name="password" required><br><br>
    <input type="submit" value="Login">
  </form>
  ';
}
?>
```

OUTPUT:

A module you have imported isn't available at the moment. It will be available soon.

99. Build a PHP program to apply the following tasks using session.

(a) Initiate a session (or pick up an existing one).

(b) Check for the existence of a pre-existing entry in $_SESSION. If not present, assume

that the session is new.

(c) Increment a counter that tracks how many times that the user has visited this page.

(d) Store the incremented counter back in $_SESSION.

(e) Provide a link back to the page itself, embedding the session ID as an argument if it is found.

(f) Using PHP, construct a program to restrict the user from uploading the same file again and again. If the same file already exists in that folder, display an appropriate error message.

```
<?php
// (a) Initiate session
session_start();

// (b) Check if counter exists
if (!isset($_SESSION['visit_count'])) {
   $_SESSION['visit_count'] = 0; // new session
}

// (c) Increment counter
$_SESSION['visit_count']++;

// (d) Store back
$visits = $_SESSION['visit_count'];

// (f) Initialize uploaded files array
if (!isset($_SESSION['uploaded_files'])) {
   $_SESSION['uploaded_files'] = [];
}
```

```php
$msg = "";

// NOTE: Online compilers don't support POST forms properly.
// So we simulate file upload by reading from CLI input instead of HTML form.
if (PHP_SAPI === 'cli') {
    // In compiler, simulate input with STDIN
    echo "Enter a file name to upload (or leave empty to skip): ";
    $filename = trim(fgets(STDIN));

    if ($filename !== "") {
        if (in_array($filename, $_SESSION['uploaded_files'])) {
            $msg = "Error: File '$filename' already uploaded!\n";
        } else {
            $_SESSION['uploaded_files'][] = $filename;
            $msg = "File '$filename' uploaded successfully!\n";
        }
    }
}
echo "=== Page Visit Counter ===\n";
echo "You have visited this page $visits times.\n";

echo "\n=== File Upload Simulation ===\n";
if ($msg) echo $msg;

if (!empty($_SESSION['uploaded_files'])) {
    echo "Files uploaded in this session:\n";
    foreach ($_SESSION['uploaded_files'] as $f) {
        echo " - $f\n";
    }
} else {
    echo "No files uploaded yet.\n";
}
```

OUTPUT:

```
Enter a file name to upload (or leave empty to skip): madhu
=== Page Visit Counter ===
You have visited this page 1 times.

=== File Upload Simulation ===
File 'madhu' uploaded successfully!
Files uploaded in this session:
 - madhu
```

100. Using PHP, construct a program to restrict the user from uploading the same file again and again.If the same file already exists in that folder, display an appropriate error message.

```
<?php
/**
 * Duplicate Upload Guard
 * - CLI (online compiler): simulates uploads by asking for filenames; prevents duplicates.
 * - Web server (XAMPP/WAMP): real <input type="file"> upload; prevents duplicates in /uploads.
 */
```

```
if (PHP_SAPI === 'cli') {
  // ---------- CLI MODE (online compiler friendly) ----------
  $index = [];

  // Optional persistence if the environment allows writing files
  $indexFile = __DIR__ . '/uploads_index.json';
  if (is_readable($indexFile)) {
    $data = json_decode(@file_get_contents($indexFile), true);
    if (is_array($data)) $index = $data;
  }

  echo "=== Duplicate Upload Guard (Simulation) ===\n";
  echo "Type file names to 'upload'. Press Enter on an empty line to finish.\n\n";

  while (true) {
    echo "Filename: ";
    $line = fgets(STDIN);
    if ($line === false) break;
    $name = trim($line);
    if ($name === '') break;

    if (isset($index[$name])) {
      echo "❌ Error: '$name' already exists. Please upload a different file.\n";
    } else {
      $index[$name] = true;
      echo "✅ Uploaded: '$name'\n";
    }
  }

  // Try to persist (ignore errors if FS is read-only)
  @file_put_contents($indexFile, json_encode($index, JSON_PRETTY_PRINT));

  echo "\nRecorded files this run:\n";
  if ($index) {
    foreach (array_keys($index) as $n) echo " - $n\n";
  } else {
    echo " (none)\n";
  }
  exit;
}

// ---------- WEB MODE (real server) ----------
session_start();
$msg = '';

if (!empty($_FILES['file']['name'])) {
  $uploadDir = __DIR__ . '/uploads/';
  if (!is_dir($uploadDir)) { @mkdir($uploadDir, 0777, true); }

  $name = basename($_FILES['file']['name']);
  $path = $uploadDir . $name;

  if (file_exists($path)) {
    $msg = "<p style='color:red'>Error: <b>{$name}</b> already exists. Please upload a
different file.</p>";
```

```
  } elseif (is_uploaded_file($_FILES['file']['tmp_name']) &&
move_uploaded_file($_FILES['file']['tmp_name'], $path)) {
    $msg = "<p style='color:green'>Uploaded <b>{$name}</b> successfully.</p>";
  } else {
    $msg = "<p style='color:red'>Upload failed.</p>";
  }
}
?>
<!doctype html>
<html>
<head><meta charset="utf-8"><title>Duplicate Upload Guard</title></head>
<body>
<h2>Duplicate Upload Guard</h2>
<form method="post" enctype="multipart/form-data">
 <input type="file" name="file" required>
 <button type="submit">Upload</button>
</form>
<?php echo $msg; ?>
</body>
</html>
```

OUTPUT:

```
=== Duplicate Upload Guard (Simulation) ===
Type file names to 'upload'. Press Enter on an empty line to finish.

Filename: MM
```

```
✅ Uploaded: 'MM'

Filename: MKMK

✅ Uploaded: 'MKMK'
```

101. Identify, how a cookie’s value can be retrieved in a web application?

```php
<?php
// Set a cookie (valid for 1 hour)
setcookie("username", "JohnDoe", time() + 3600, "/");

// Retrieve the cookie value
if(isset($_COOKIE["username"])) {
   echo "Cookie value is: " . $_COOKIE["username"];
} else {
   echo "Cookie is not set yet!";
}
?>
```

OUTPUT:

```
Cookie is not set yet!
```

102. Develop a PHP program to use session variables and start a session.

```php
<?php
// Start the session
session_start();

// Set session variable
if(!isset($_SESSION["username"])) {
   $_SESSION["username"] = "JohnDoe";
   echo "Session started and variable set!<br>";
} else {
    echo "Session already active.<br>";
}

// Retrieve session variable
echo "Username stored in session is: " . $_SESSION["username"];
?>
```

OUTPUT:

```
Session started and variable set!<br>Username stored in session is: JohnDoe
```

103. Create a very simple pageview counter by using 'isset' to check if the pageview variable has already been created.

```
<?php
// Start the session
session_start();

// Check if 'views' is already set
if(isset($_SESSION['views'])) {
    $_SESSION['views'] = $_SESSION['views'] + 1;   // Increment counter
} else {
    $_SESSION['views'] = 1;   // First visit
}

// Display the counter
echo "You have visited this page " . $_SESSION['views'] . " times.";
?>
```

OUTPUT:

You have visited this page 1 times.

104. Formulate a PHP program to open a text file and print the nth line in the text file if then'th line does not exist print 'no data'.

```
<?php

// File name

$filename = "sample.txt";


// Line number to fetch (example: 3rd line)

$n = 3;


// Check if file exists

if(file_exists($filename)) {

  $file = file($filename); // Reads entire file into an array (each line = array element)


  if(isset($file[$n - 1])) {

    echo "Line $n: " . htmlspecialchars($file[$n - 1]);

  } else {

    echo "no data";

  }

} else {

  echo "File not found.";

}

?>
```

OUTPUT:

```
File not found.
```

105. Write a PHP function to get the start and end date of a week (by week number) of a particular

year. Sample week and year: 12, 2014.

```
<?php

function getStartAndEndDate($week, $year) {

   // Create a DateTime object with ISO week format

   $dto = new DateTime();

   $dto->setISODate($year, $week);


   // Start of the week (Monday)

   $start = $dto->format('Y-m-d');


   // End of the week (Sunday)

   $dto->modify('+6 days');

   $end = $dto->format('Y-m-d');


   return array($start, $end);

}


// Example usage

list($startDate, $endDate) = getStartAndEndDate(12, 2014);


echo "Week 12 of 2014 starts on: " . $startDate . "<br>";

echo "Week 12 of 2014 ends on: " . $endDate;

?>
```

OUTPUT:

```
Week 12 of 2014 starts on: 2014-03-17<br>Week 12 of 2014 ends on: 2014-03-23
```