UNIT-IV

1. Introduction and basic syntax of PHP

Ans: PHP is a simple server-side scripting language with roots in C and Perl but resembles Java in its syntax, commonly used for web development to create dynamic and interactive web applications.

Incorporating PHP within HTML:
- PHP documents typically have the extension .php and are automatically processed by the web server.
- Some developers configure web servers to parse .htm or .html files as PHP for security reasons.
- PHP can output HTML, making it easy to create web content.

Calling the PHP Parser:
- PHP code is enclosed within <?php and ?> tags.
- These tags can wrap entire documents or just small PHP fragments as needed.
- The choice of how to use these tags depends on coding style and preference.
- There's an alternative opening and closing syntax, <? and ?>, but it's discouraged and deprecated.

The Structure of PHP:
- PHP supports single-line comments with // and multi-line comments with /* */.
- Single-line comments are useful for temporarily disabling code or adding explanations.
- Multi-line comments are used to comment out larger sections of code.
- Avoid nesting multi-line comments to prevent interpreter errors.

Basic Syntax:
- **Semicolons:** PHP commands must end with a semicolon (;).
- **$ Symbol:** a dollar sign ($) must precede all variable names.
- **Whitespace and Formatting:** PHP doesn't enforce strict code indentation or layout rules but sensible use of whitespace and comprehensive commenting is encouraged for code readability and maintenance.

Ex:

```php
<?php
// Variable assignment
$mycounter = 1;
$mystring = "Hello";
$myarray = array("One", "Two", "Three");

// Displaying Results
echo $mycounter."\n";
echo $mystring."\n";
echo $myarray[0];
?>
```

Output:

```
PS C:\Users\DELL\myproject> php index.php
1
Hello
One
```

DataTypes and Operators:

```php
<?php

// Arithmetic Operators
$a = 10;
$b = 5;
$sum = $a + $b;
```

```php
$subtract = $a - $b;
$multiply = $a * $b;
$divide = $a / $b;
$modulus = $a % $b;

// Comparison Operators
$num1 = 10;
$num2 = 5;
$equalResult = ($num1 == $num2);
$notEqualResult = ($num1 != $num2);
$greaterThanResult = ($num1 > $num2);
$lessThanResult = ($num1 < $num2);
$greaterThanOrEqualResult = ($num1 >= $num2);
$lessThanOrEqualResult = ($num1 <= $num2);

// Logical Operators
$x = true;
$y = false;
$logicalAndResult = ($x && $y);
$logicalOrResult = ($x || $y);
$logicalNotX = !$x;
$logicalNotY = !$y;

// Assignment Operators
$assignA = 10;
$assignB = 5;
$assignA += $assignB;
$assignA -= $assignB;
$assignA *= $assignB;
$assignA /= $assignB;
$assignA %= $assignB;

// Concatenation Operator (String)
$str1 = "Hello, ";
$str2 = "world!";
$concatResult = $str1 . $str2;

// Data Types
$integer = 42;
$float = 3.14;
$string = "John";
$boolean = true;
$array = array("red", "green", "blue");

// Displaying Results
echo "\$sum = $sum\n";
echo "\$subtract = $subtract\n";
echo "\$multiply = $multiply\n";
echo "\$divide = $divide\n";
echo "\$modulus = $modulus\n";
echo "\$equalResult = ".($equalResult ? "true" : "false")."\n";
echo "\$notEqualResult = ".($notEqualResult ? "true" : "false")."\n";
```

```php
    echo "\$greaterThanResult = ".($greaterThanResult ? "true" :
"false")."\n";
    echo "\$lessThanResult = ".($lessThanResult ? "true" : "false")."\n";
    echo "\$greaterThanOrEqualResult = ".($greaterThanOrEqualResult ?
"true" : "false")."\n";
    echo "\$lessThanOrEqualResult = ".($lessThanOrEqualResult ? "true" :
"false")."\n";
    echo "\$logicalAndResult = ".($logicalAndResult ? "true" :
"false")."\n";
    echo "\$logicalOrResult = ".($logicalOrResult ? "true" :
"false")."\n";
    echo "\$logicalNotX = ".($logicalNotX ? "true" : "false")."\n";
    echo "\$logicalNotY = ".($logicalNotY ? "true" : "false")."\n";
    echo "\$assignA = $assignA\n";
    echo "\$concatResult = $concatResult\n";
    echo "\$integer = $integer\n";
    echo "\$float = $float\n";
    echo "\$string = $string\n";
    echo "\$boolean = ".($boolean ? "true" : "false")."\n";
    echo "\$array[0] = $array[0]\n";

    ?>
```

Ouput:

```
PS D:\MYDATABASE\VINNY\DOCUMENTS\VRSEC\STUFF\Full Stack Honors\SEM-3\basic_project\basic_app> php index.php
$sum = 15
$subtract = 5
$multiply = 50
$divide = 2
$modulus = 0
$equalResult = false
$notEqualResult = true
$greaterThanResult = true
$lessThanResult = false
$greaterThanOrEqualResult = true
$lessThanOrEqualResult = false
$logicalAndResult = false
$logicalOrResult = true
$logicalNotX = false
$logicalNotY = true
$assignA = 0
$concatResult = Hello, world!
$integer = 42
$float = 3.14
$string = John
$boolean = true
$array[0] = red
```

2. Decision and Looping with examples.
Ans:
```php
<?php

// Decision Statements

// If Statement
$condition = true;
if ($condition) {
    echo "This is an if statement.\n";
}
```

```php
// If-Else Statement
$age = 20;
if ($age >= 18) {
    echo "You are an adult.\n";
} else {
    echo "You are not an adult.\n";
}

// If-Elseif-Else Statement
$grade = "B";
if ($grade == "A") {
    echo "Excellent!\n";
} elseif ($grade == "B") {
    echo "Good!\n";
} else {
    echo "Needs improvement.\n";
}

// Switch Statement
$day = "Monday";
switch ($day) {
    case "Monday":
        echo "It's Monday.\n";
        break;
    case "Tuesday":
        echo "It's Tuesday.\n";
        break;
    default:
        echo "It's some other day.\n";
}

// Looping Statements

// While Loop
$num = 1;
while ($num <= 5) {
    echo "While loop: $num\n";
    $num++;
}

// Do-While Loop
$num = 1;
do {
    echo "Do-While loop: $num\n";
    $num++;
} while ($num <= 5);

// For Loop
for ($i = 1; $i <= 5; $i++) {
    echo "For loop: $i\n";
}
```
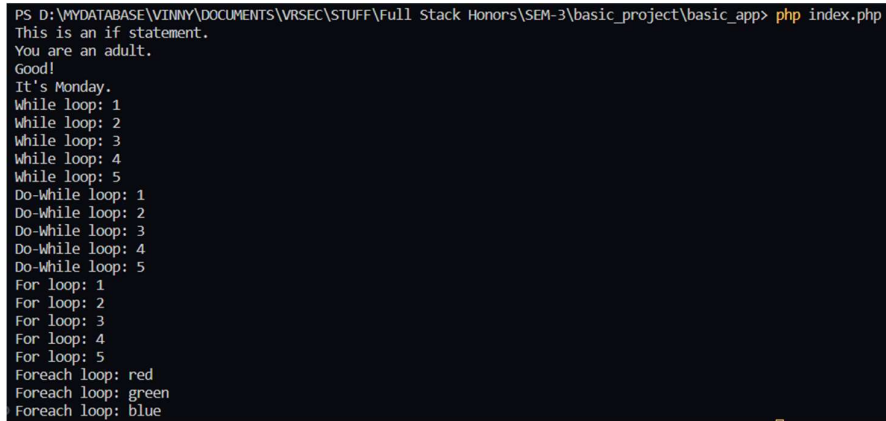
```php
// Foreach Loop (for arrays)
$colors = array("red", "green", "blue");
foreach ($colors as $color) {
    echo "Foreach loop: $color\n";
}

?>
```
Output:

3. PHP and HTML
Ans:
Incorporating php with html:

```php
<!DOCTYPE html>
<html>
<head>
    <title>PHP within HTML</title>
</head>
<body>

<?php
// PHP within HTML
$firstName = "John";
$lastName = "Doe";
$age = 30;
?>

<h1>Welcome, <?php echo $firstName . ' ' . $lastName; ?>!</h1>
<p>You are <?php echo $age; ?> years old.</p>

<?php
// PHP Condition within HTML
$isLoggedIn = true;
?>

<?php if ($isLoggedIn): ?>
    <p>You are currently logged in.</p>
<?php else: ?>
    <p>You are not logged in.</p>
<?php endif; ?>

<!-- PHP Loop within HTML -->
```
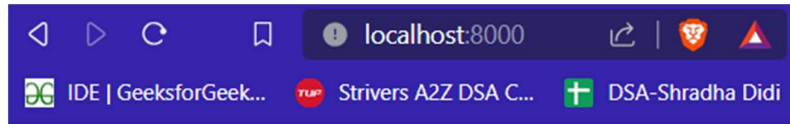
```html
<ul>
    <?php for ($i = 1; $i <= 5; $i++): ?>
        <li>Item <?php echo $i; ?></li>
    <?php endfor; ?>
</ul>

</body>
</html>
```
Output:

Welcome, John Doe!

You are 30 years old.

You are currently logged in.

- Item 1
- Item 2
- Item 3
- Item 4
- Item 5

4. **Arrays:**

Ans:
```php
<?php
// Creating an indexed array
$fruits = array("apple", "banana", "cherry", "date");

// Accessing elements by index
echo "Indexed Array:\n";
echo "The first fruit is: " . $fruits[0] . "\n";
echo "The second fruit is: " . $fruits[1] . "\n";

// Adding elements to an indexed array
$fruits[] = "grape";
$fruits[] = "kiwi";
echo "Newly added fruits: " . end($fruits) . " and " . prev($fruits) . "\n";

// Associative array
$person = array(
    "first_name" => "John",
    "last_name" => "Doe",
    "age" => 30,
```

```php
);

echo "\nAssociative Array:\n";
echo "First Name: " . $person["first_name"] . "\n";
echo "Last Name: " . $person["last_name"] . "\n";
echo "Age: " . $person["age"] . "\n";

// Multidimensional array
$students = array(
    array("name" => "Alice", "grade" => "A"),
    array("name" => "Bob", "grade" => "B"),
    array("name" => "Charlie", "grade" => "C"),
);

echo "\nMultidimensional Array:\n";
foreach ($students as $student) {
    echo "Student: " . $student["name"] . ", Grade: " .
$student["grade"] . "\n";
}

// Using array functions
$numbers = array(5, 3, 8, 2, 1, 7);

echo "\nArray Functions:\n";
echo "Original Array: " . implode(", ", $numbers) . "\n";

// Sorting
sort($numbers);
echo "Sorted Array (Ascending): " . implode(", ", $numbers) . "\n";

rsort($numbers);
echo "Sorted Array (Descending): " . implode(", ", $numbers) . "\n";

// Counting elements
$numElements = count($numbers);
echo "Number of Elements in the Array: " . $numElements . "\n";

// Searching
$searchValue = 3;
if (in_array($searchValue, $numbers)) {
    echo "The value $searchValue is found in the array.\n";
} else {
    echo "The value $searchValue is not found in the array.\n";
}

// Removing duplicates
$uniqueNumbers = array_unique($numbers);
echo "Array with Duplicates Removed: " . implode(", ", $uniqueNumbers)
. "\n";

// Shuffle
shuffle($numbers);
echo "Shuffled Array: " . implode(", ", $numbers) . "\n";
```

```php
// Explode
$commaSeparated = "apple,banana,cherry,date";
$explodedArray = explode(",", $commaSeparated);
echo "Exploded Array: " . implode(", ", $explodedArray) . "\n";

// Extract
extract($person);
echo "Extracted Variables - First Name: $first_name, Last Name:
$last_name, Age: $age\n";

// Compact
$city = "New York";
$state = "NY";
$country = "USA";
$address = compact('city', 'state', 'country');
print_r($address);

// Reset
reset($fruits);
echo "After reset, the array pointer is at: " . current($fruits) .
"\n";
?>
```

Output:

```
PS C:\Users\DELL\myproject> php index.php
Indexed Array:
The first fruit is: apple
The second fruit is: banana
Newly added fruits: kiwi and grape

Associative Array:
First Name: John
Last Name: Doe
Age: 30

Multidimensional Array:
Student: Alice, Grade: A
Student: Bob, Grade: B
Student: Charlie, Grade: C

Array Functions:
Original Array: 5, 3, 8, 2, 1, 7
Sorted Array (Ascending): 1, 2, 3, 5, 7, 8
Sorted Array (Descending): 8, 7, 5, 3, 2, 1
Number of Elements in the Array: 6
The value 3 is found in the array.
Array with Duplicates Removed: 8, 7, 5, 3, 2, 1
Shuffled Array: 5, 8, 7, 3, 1, 2
Exploded Array: apple, banana, cherry, date
Extracted Variables - First Name: John, Last Name: Doe, Age: 30
Array
(
    [city] => New York
    [state] => NY
    [country] => USA
)
After reset, the array pointer is at: apple
```

5. Functions
Ans:
```php
<?php
// Function without parameters
function sayHello() {
    echo "Hello, World!\n";
```

```php
}

sayHello(); // Call the function

// Function with parameters
function add($a, $b) {
    $result = $a + $b;
    return $result;
}

$sum = add(5, 3); // Call the function and store the result in $sum
echo "The sum is: $sum\n";

// Function with default parameter value
function greet($name = "Guest") {
    echo "Hello, $name!\n";
}

greet(); // Call with default parameter
greet("John"); // Call with a specific parameter

// Function with variable-length argument list
function calculateSum(...$numbers) {
    $sum = array_sum($numbers);
    return $sum;
}

$total = calculateSum(1, 2, 3, 4, 5);
echo "Total sum is: $total\n";

// Function with a return statement
function isEven($number) {
    if ($number % 2 == 0) {
        return true;
    } else {
        return false;
    }
}

$checkNumber = 7;
if (isEven($checkNumber)) {
    echo "$checkNumber is even.\n";
} else {
    echo "$checkNumber is odd.\n";
}

// Function with a global variable
$globalVar = 10;

function accessGlobalVar() {
    global $globalVar;
    echo "Global variable value is: $globalVar\n";
}
```

```php
    accessGlobalVar();

    // Recursive function
    function factorial($n) {
        if ($n <= 1) {
            return 1;
        } else {
            return $n * factorial($n - 1);
        }
    }

    $fact = factorial(5);
    echo "Factorial of 5 is: $fact\n";
    ?>
```
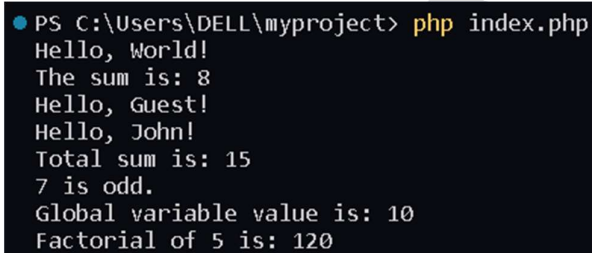**Output:**

```
PS C:\Users\DELL\myproject> php index.php
Hello, World!
The sum is: 8
Hello, Guest!
Hello, John!
Total sum is: 15
7 is odd.
Global variable value is: 10
Factorial of 5 is: 120
```

6. Browser control and detection:
```php
<?php
// Basic browser detection and control

// Get the user-agent string
$userAgent = $_SERVER['HTTP_USER_AGENT'];

// Check if the user-agent string contains specific keywords to
identify browsers
$isChrome = strpos($userAgent, 'Chrome') !== false;
$isFirefox = strpos($userAgent, 'Firefox') !== false;
$isIE = strpos($userAgent, 'MSIE') !== false || strpos($userAgent,
'Trident/') !== false;

// Detect the user's browser and perform actions accordingly
if ($isChrome) {
    $message = "You are using Google Chrome.";
} elseif ($isFirefox) {
    $message = "You are using Mozilla Firefox.";
} elseif ($isIE) {
    $message = "You are using Internet Explorer.";
} else {
    $message = "Your browser is not recognized.";
}

// Set content type to HTML
header('Content-Type: text/html; charset=UTF-8');

// Output the message
```

```php
    echo "<html><head><title>Browser Detection</title></head><body>";
    echo "<h1>Browser Detection</h1>";
    echo "<p>$message</p>";

    // Redirect users based on their browser
    if ($isIE) {
        echo "<p>Sorry, this website does not support Internet Explorer.
Please use a different browser.</p>";
        echo '<script>setTimeout(function() { window.location.href =
"https://www.mozilla.org/en-US/firefox/new/"; }, 5000);</script>';
    } else {
        echo "<p>Welcome to our website! You can continue browsing.</p>";
    }

    echo "</body></html>";
    ?>
```

Output:

Browser Detection

You are using Google Chrome.

Welcome to our website! You can continue browsing.

7. String
Ans:

```php
<?php
// String Basics

// Creating strings
$string1 = "Hello, ";
$string2 = 'World!';

// Concatenation
$concatenatedString = $string1 . $string2;
echo "Concatenated String: $concatenatedString\n";

// String Length
$length = strlen($concatenatedString);
echo "String Length: $length\n";

// Accessing Characters
$firstChar = $concatenatedString[0];
echo "First Character: $firstChar\n";

// Substrings
$substring = substr($concatenatedString, 0, 5);
echo "Substring: $substring\n";

// String Functions

// Uppercase and Lowercase
$uppercase = strtoupper($concatenatedString);
$lowercase = strtolower($concatenatedString);
```

```php
echo "Uppercase: $uppercase\n";
echo "Lowercase: $lowercase\n";

// Replace
$replacedString = str_replace("Hello", "Hi", $concatenatedString);
echo "Replaced String: $replacedString\n";

// Find Position
$position = strpos($concatenatedString, "World");
echo "Position of 'World': $position\n";

// Trim
$whitespaceString = "   Trim me!   ";
$trimmedString = trim($whitespaceString);
echo "Trimmed String: '$trimmedString'\n";

// Explode and Implode
$colors = "red,green,blue";
$colorArray = explode(",", $colors);
echo "Exploded Array: ";
print_r($colorArray);

$reconstructedString = implode(" | ", $colorArray);
echo "\nReconstructed String: $reconstructedString\n";

// String Interpolation
$variable = "PHP";
$interpolatedString = "I love $variable!";
echo "Interpolated String: $interpolatedString\n";

// Escaping Characters
$escapedString = "She said, \"Hello!\"";
echo "Escaped String: $escapedString\n";

// Multiline Strings
$multilineString = <<<EOD
This is a
multiline
string.
EOD;
echo "Multiline String:\n$multilineString\n";

// String Encoding
$utf8String = "UTF-8 Encoding: üöä";
$encodedString = utf8_encode($utf8String);
$decodedString = utf8_decode($encodedString);
echo "Original String: $utf8String\n";
echo "Encoded String: $encodedString\n";
echo "Decoded String: $decodedString\n";
?>
```

Output:

```
PS C:\Users\DELL\myproject> php index.php
Concatenated String: Hello, World!
String Length: 13
First Character: H
Substring: Hello
Uppercase: HELLO, WORLD!
Lowercase: hello, world!
Replaced String: Hi, World!
Position of 'World': 7
Trimmed String: 'Trim me!'
Exploded Array: Array
(
    [0] => red
    [1] => green
    [2] => blue
)

Reconstructed String: red | green | blue
Interpolated String: I love PHP!
Escaped String: She said, "Hello!"
Multiline String:
This is a
multiline
string.
Original String: UTF-8 Encoding: üöä
Encoded String: UTF-8 Encoding: Ã¼Ã¶Ã¤
Decoded String: UTF-8 Encoding: üöä
```

8. Form processing

Ans:

```
<!DOCTYPE html>
<html>
<head>
    <title>Simple Form Processing</title>
</head>
<body>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST["name"];
    $email = $_POST["email"];
    $message = $_POST["message"];

    echo "<h2>Form Submitted Successfully</h2>";
    echo "<p>Name: $name</p>";
    echo "<p>Email: $email</p>";
    echo "<p>Message: $message</p>";
}
?>

<h1>Contact Us</h1>
<form method="post" action="<?php echo $_SERVER['PHP_SELF']; ?>">
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required><br><br>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required><br><br>

    <label for="message">Message:</label><br>
    <textarea id="message" name="message" rows="4" cols="50"
required></textarea><br><br>
```

```
        <input type="submit" value="Submit">
</form>

</body>
</html>
```
Output:

Form Submitted Successfully

Name: Andrew Garfield

Email: ag2012@gmail.com

Message: But those are the best kind :)

Contact Us

Name: [   ]

Email: [   ]

Message:

[   ]

[Submit]

9. File
Ans:
```php
<?php
// File Basics

// File path
$filePath = "example.txt";

// Opening a file for writing
$file = fopen($filePath, "w");

// Writing to a file
fwrite($file, "Hello, World!\n");
fwrite($file, "This is a PHP file.\n");

// Closing the file
fclose($file);

// Opening a file for reading
$file = fopen($filePath, "r");

// Reading from a file and outputting its content
while (!feof($file)) {
    echo fgets($file);
}
```
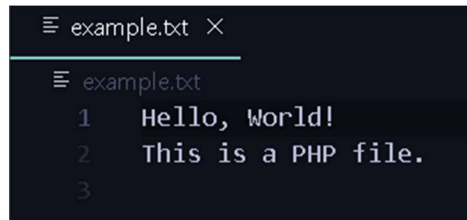
```php
// Closing the file
fclose($file);
?>
```
Output:

10. Cookies and Sessions
Ans:
```php
<?php
// Cookies and Sessions Basics

// Start a session
session_start();

// Set a session variable
$_SESSION['username'] = 'JohnDoe';

// Set a cookie
setcookie('user', 'Alice', time() + 3600, '/');

// Display session value
echo "Session Username: " . $_SESSION['username'] . "<br>";

// Check if the cookie is set and display its value
if (isset($_COOKIE['user'])) {
    echo "Cookie User: " . $_COOKIE['user'] . "<br>";
} else {
    echo "Cookie User is not set.<br>";
}

// Modify the session and cookie values
$_SESSION['username'] = 'JaneSmith';
setcookie('user', 'Bob', time() + 3600, '/');

// Display modified values
echo "Modified Session Username: " . $_SESSION['username'] . "<br>";

// Check if the cookie is set and display its modified value
if (isset($_COOKIE['user'])) {
    echo "Modified Cookie User: " . $_COOKIE['user'] . "<br>";
} else {
    echo "Modified Cookie User is not set.<br>";
}

// End the session
session_destroy();

// Unset the cookie
```
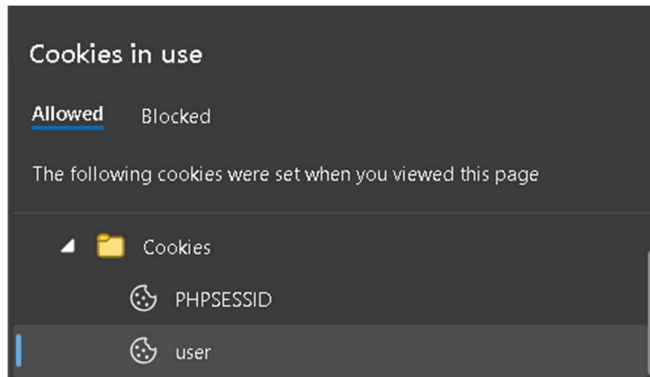
```
setcookie('user', '', time() - 3600, '/');
?>
```
Output:

Session Username: JohnDoe
Cookie User is not set.
Modified Session Username: JaneSmith
Modified Cookie User is not set.

11. Object Oriented Programming with PHP
Ans:
```php
<?php
// Parent Class (Base Class)
class Person {
    // Public properties
    public $name;
    public $age;

    // Constructor method
    public function __construct($name, $age) {
        $this->name = $name;
        $this->age = $age;
    }

    // Public method
    public function displayInfo() {
        echo "Name: $this->name, Age: $this->age\n";
    }

    // Final method (cannot be overridden in child classes)
    final public function introduce() {
        echo "Hello, I'm $this->name!\n";
    }

    // Static method
    public static function sayHello() {
        echo "Hello from the Person class!\n";
    }
}
```

```php
// Child Class (Derived Class)
class Student extends Person {
    // Public property specific to the Student class
    public $studentId;

    // Constructor method specific to the Student class
    public function __construct($name, $age, $studentId) {
        // Call the parent class constructor
        parent::__construct($name, $age);
        $this->studentId = $studentId;
    }

    // Public method in the child class
    public function study() {
        echo "$this->name is studying.\n";
    }

    // Static method in the child class
    public static function sayHello() {
        echo "Hello from the Student class!\n";
    }
}

// Create objects of the Student class
$student1 = new Student("John Doe", 20, "S12345");
$student2 = new Student("Jane Smith", 22, "S67890");

// Access public properties and methods
echo "Student 1 Info: ";
$student1->displayInfo();
echo "Student 2 Info: ";
$student2->displayInfo();

// Access a public method in the child class
$student1->study();

// Access a final method from the parent class
echo "Student 2 Introduction: ";
$student2->introduce();

// Access a static method
Person::sayHello(); // Calls the static method in the Parent class
Student::sayHello(); // Calls the static method in the Child class
?>
```
Output:

```
PS C:\Users\DELL\myproject> php index.php
Student 1 Info: Name: John Doe, Age: 20
Student 2 Info: Name: Jane Smith, Age: 22
John Doe is studying.
Student 2 Introduction: Hello, I'm Jane Smith!
Hello from the Person class!
Hello from the Student class!
```

12. MySQL using PHP
Ans:

```php
<?php
// Database Connection and Operations

// Database Configuration
$servername = "localhost";
$username = "root";
$password = "root";

// Create a connection
$conn = new mysqli($servername, $username, $password);

// Check the connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Create a new database
$database = "example_db";
$sqlCreateDatabase = "CREATE DATABASE IF NOT EXISTS $database";
if ($conn->query($sqlCreateDatabase) === TRUE) {
    echo "Database created or exists: $database\n";
} else {
    die("Error creating database: " . $conn->error);
}

// List databases
$databases = $conn->query("SHOW DATABASES");
if ($databases->num_rows > 0) {
    echo "\nDatabases:\n";
    while ($row = $databases->fetch_assoc()) {
        echo $row["Database"] . "\n";
    }
} else {
    echo "\nNo databases found.\n";
}

// Select the database
$conn->select_db($database);

// Create a table
$sqlCreateTable = "CREATE TABLE IF NOT EXISTS users (
    id INT AUTO_INCREMENT PRIMARY KEY,
    username VARCHAR(30),
    email VARCHAR(50)
)";
if ($conn->query($sqlCreateTable) === TRUE) {
    echo "\nTable 'users' created or exists\n";
} else {
    die("Error creating table: " . $conn->error);
}
```

```php
    // List tables
    $tables = $conn->query("SHOW TABLES");
    if ($tables->num_rows > 0) {
        echo "\nTables in $database:\n";
        while ($row = $tables->fetch_assoc()) {
            echo $row["Tables_in_$database"] . "\n";
        }
    } else {
        echo "\nNo tables found in $database.\n";
    }

    // Insert data into the table
    $sqlInsertData = "INSERT INTO users (username, email) VALUES
    ('john_doe', 'john@example.com')";
    if ($conn->query($sqlInsertData) === TRUE) {
        echo "\nData inserted successfully\n";
    } else {
        die("\nError inserting data: " . $conn->error);
    }

    // Alter the table (Add a new column)
    $sqlAlterTable = "ALTER TABLE users ADD COLUMN age INT";
    if ($conn->query($sqlAlterTable) === TRUE) {
        echo "\nTable 'users' altered successfully\n";
    } else {
        die("\nError altering table: " . $conn->error);
    }

    // Run a sample query (Select data)
    $sqlSelectData = "SELECT * FROM users";
    $result = $conn->query($sqlSelectData);
    if ($result->num_rows > 0) {
        echo "\nUsers:\n";
        while ($row = $result->fetch_assoc()) {
            echo "ID: " . $row["id"] . ", Username: " . $row["username"] .
    ", Email: " . $row["email"] . "\n";
        }
    } else {
        echo "\nNo users found.\n";
    }

    // Delete data from the table
    $sqlDeleteData = "DELETE FROM users WHERE username = 'john_doe'";
    if ($conn->query($sqlDeleteData) === TRUE) {
        echo "\nData deleted successfully\n";
    } else {
        die("\nError deleting data: " . $conn->error);
    }

    // Delete the table
    $sqlDeleteTable = "DROP TABLE IF EXISTS users";
    if ($conn->query($sqlDeleteTable) === TRUE) {
        echo "\nTable 'users' deleted or does not exist\n";
```

```php
} else {
    die("\nError deleting table: " . $conn->error);
}

// Delete the database
$sqlDeleteDatabase = "DROP DATABASE IF EXISTS $database";
if ($conn->query($sqlDeleteDatabase) === TRUE) {
    echo "\nDatabase '$database' deleted or does not exist\n";
} else {
    die("\nError deleting database: " . $conn->error);
}

// Close the connection
$conn->close();
?>
```
Output:

```
PS C:\Users\DELL\myproject> php index.php
Database created or exists: example_db

Databases:
class
db
exam
example_db
information_schema
lab1
mydb
mysql
performance_schema
sakila
sys
temp
world
xyz

Table 'users' created or exists

Tables in example_db:
users

Data inserted successfully

Table 'users' altered successfully

Users:
ID: 1, Username: john_doe, Email: john@example.com

Data deleted successfully

Table 'users' deleted or does not exist

Database 'example_db' deleted or does not exist
```
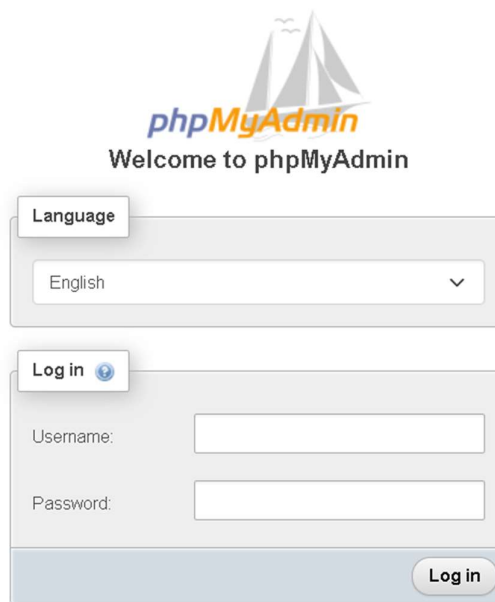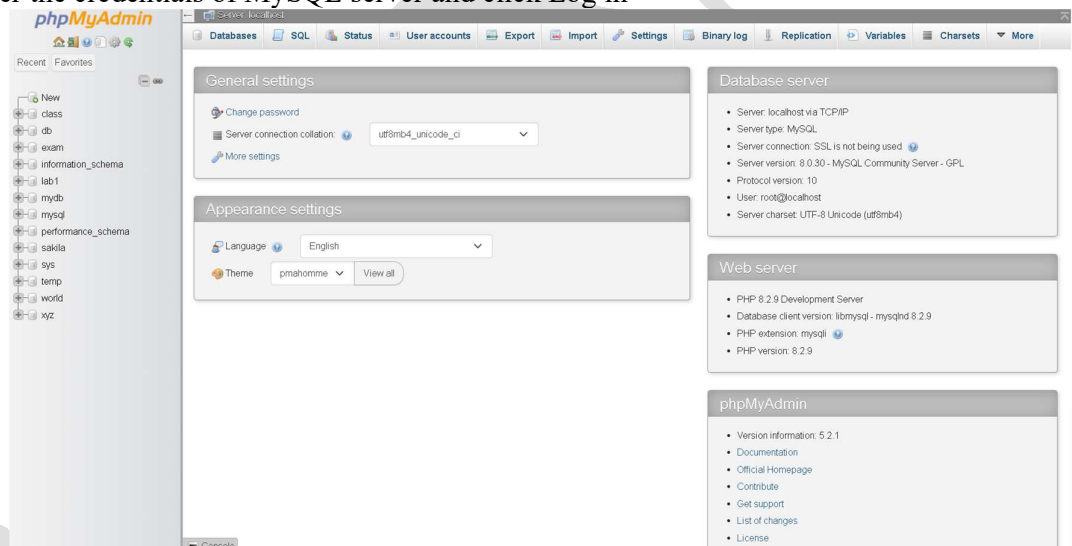
13. PHP myadmin and data bugs
Ans: phpMyAdmin is a popular web-based database management tool for MySQL database.
- It provides a user-friendly interface for performing tasks like creating, modifying, and querying databases, as well as managing database users and privileges.

- phpMyAdmin is widely used by developers and database administrators to simplify database management tasks through a web browser.
- Open a web browser and navigate to **http://localhost:8000/phpMyAdmin/index.php**

- Enter the credentials of MySQL server and click Log in

- Following are some common data-related issues or "data bugs" that can occur when using phpMyAdmin:
  - **Data Entry Errors:** When users enter data that is inaccurate or doesn't match the intended format, leading to data errors.
  - **Data Duplication**: Unintended duplication of records or rows in database tables.
  - **Data Validation Failures:** Data that does not meet specified validation criteria causing issues like invalid phone numbers or emails.
  - **Data Integrity Violations:** When relationships between data are broken, causing inconsistencies or errors.
  - **Data Loss:** Accidental deletion or overwriting of important data, resulting in permanent loss of information.
  - **Data Security Breaches:** Unauthorized access or exposure of sensitive data, which can lead to data leaks or privacy breaches.