Experiment No. 1

Title: Simple PHP programs using basic programming Constructs

Batch: B-3 **Roll No.: 16010422234** **Experiment No.:1**

Aim: Write PHP programs using basic programming constructs such as data types, control structures, String functions , date and time functions and Math functions

---

Resources needed: Windows OS, Web Browser, Editor, XAMPP Server

---

Pre Lab/ Prior Concepts:
Students should have prior knowledge of HTML/CSS/Basic Programming.

Theory:

PHP

1. PHP is a server side scripting language.
2. It can be used to develop Static websites or Dynamic websites or Web applications.
3. PHP stands forHypertext Preprocessor, which earlier stood for Personal Home Pages.
4. PHP scripts can only be interpreted on a server that has PHP installed.
5. The client computers accessing the PHP scripts require a web browser only.
6. A PHP file contains PHP tags and ends with the extension ".php".
7. PHP code may be embedded into HTML code, or it can be used in combination with various web template systems, web content management systems and web frameworks.

PHP supports the following data types:

- String
- Integer
- Float (floating point numbers - also called double)
- Boolean
- Array
- Object
- NULL
- Resource

PHP String

A string is a sequence of characters, like "Hello world!".

A string can be any text inside quotes. You can use single or double quotes:

```
<?php
$x = "Hello world!";
$y = 'Hello world!';

echo $x;
echo
"<br>";
echo $y;
```

```
?>
```

PHP has a set of math functions that allows you to perform mathematical tasks on numbers.
For Example : The pi() function returns the value of PI
The abs() function returns the absolute (positive) value of a number
Other Examples: min(), max() etc.

PHP Conditional Statements
if statement - executes some code if one condition is true
if...else statement - executes some code if a condition is true and another code if that condition is false
if...elseif...else statement - executes different codes for more than two conditions
switch statement - selects one of many blocks of code to be executed

Sample Code: if else and if elseif else example

if (condition) {
code to be executed if condition is true;
} else {
code to be executed if condition is false;
}

```
<?php
$t = date("H");
if ($t < "10") {
echo "Have a good morning!";
} elseif ($t < "20") { echo "Have a good day!";
} else {
echo "Have a good night!";
}
?>
```

Similarly all the conditional constructs such as for loop, switch case etc are available in PHP

Echo and Print

echo and print are more or less the same. 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.

We can use html tags in php e.g.
, <b> used in the above example.

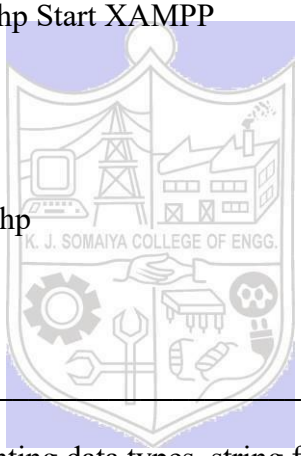
Procedure:

How to Run a PHP File in XAMPP?

Step 1 : First Create PHP script using any editor like notepad, notepad++ etc.
<?php echo "Welcome to the world of PHP."; ?>

Step 2 : Save file as following... firstProg.php
In C:xampp/htdocs/myproject/firstProg.php Start XAMPP
Apache server (first time only)

Step 3 : Run the PHP script
Open Your browser and write in
Type url : localhost/myproject/firstProg.php



Output:
Welcome to the world of PHP.

Activity: Write a PHP program implementing data types, string functions, date and time functions, Math functions, control structures.

Output (code with result snapshot):-

Code:

EXP1.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Contact Manager</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f7f7f7;
            margin: 0;
            padding: 20px;
            box-sizing: border-box;
        }
        h2, h3 {
            color: #333;
        }
        form {
            margin-bottom: 20px;
        }
        label {
```

```
            display: inline-block;
            width: 120px;
            margin-bottom: 10px;
        }
        input[type="text"],
        input[type="email"],
        input[type="date"] {
            width: 200px;
            padding: 5px;
            margin-bottom: 10px;
        }
        input[type="submit"] {
            padding: 5px 10px;
            background-color: #337ab7;
            color: white;
            border: none;
            cursor: pointer;
        }
        .table-container {
            margin-top: 20px;
        }
        table {
            width: 100%;
            border-collapse: collapse;
        }
        table, th, td {
            border: 1px solid #ddd;
        }
        th, td {
            padding: 8px;
            text-align: left;
        }
        th {
            background-color: #337ab7;
            color: white;
        }
    </style>
</head>
<body>
    <h2>Contact Manager</h2>
    <form method="post" action="">
        <label for="name">Name:</label>
```

```
        <input type="text" id="name" name="name" required><br>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email" required><br>
        <label for="phone">Phone:</label>
        <input type="text" id="phone" name="phone" required><br>
        <label for="birthdate">Birthdate:</label>
        <input type="date" id="birthdate" name="birthdate" required><br>
        <label for="color">Favorite Color:</label>
        <input type="text" id="color" name="color" required><br>
        <input type="submit" value="Add Contact">
    </form>
    <h2>Contact List</h2>
    <div class="table-container">
        <?php
        function calculate_age($birthdate) {
            $birth_date = new DateTime($birthdate);
            $current_date = new DateTime();
            $age = $current_date->diff($birth_date)->y;
            return $age;
        }

        if ($_SERVER['REQUEST_METHOD'] == 'POST') {
            $name = $_POST['name'];
            $email = $_POST['email'];
            $phone = $_POST['phone'];
            $birthdate = $_POST['birthdate'];
            $color = $_POST['color'];
            $name = ucwords(strtolower(trim($name)));
            $email = strtolower(trim($email));
            $file = fopen("contacts.txt", "a");
            fwrite($file, "$name,$email,$phone,$birthdate,$color\n");
            fclose($file);
        }

        function display_contacts_and_average_age() {
            if (file_exists("contacts.txt")) {
                $file = fopen("contacts.txt", "r");
                echo "<table>
                        <tr>
                        <th>Name</th>
                        <th>Email</th>
                        <th>Phone</th>
```

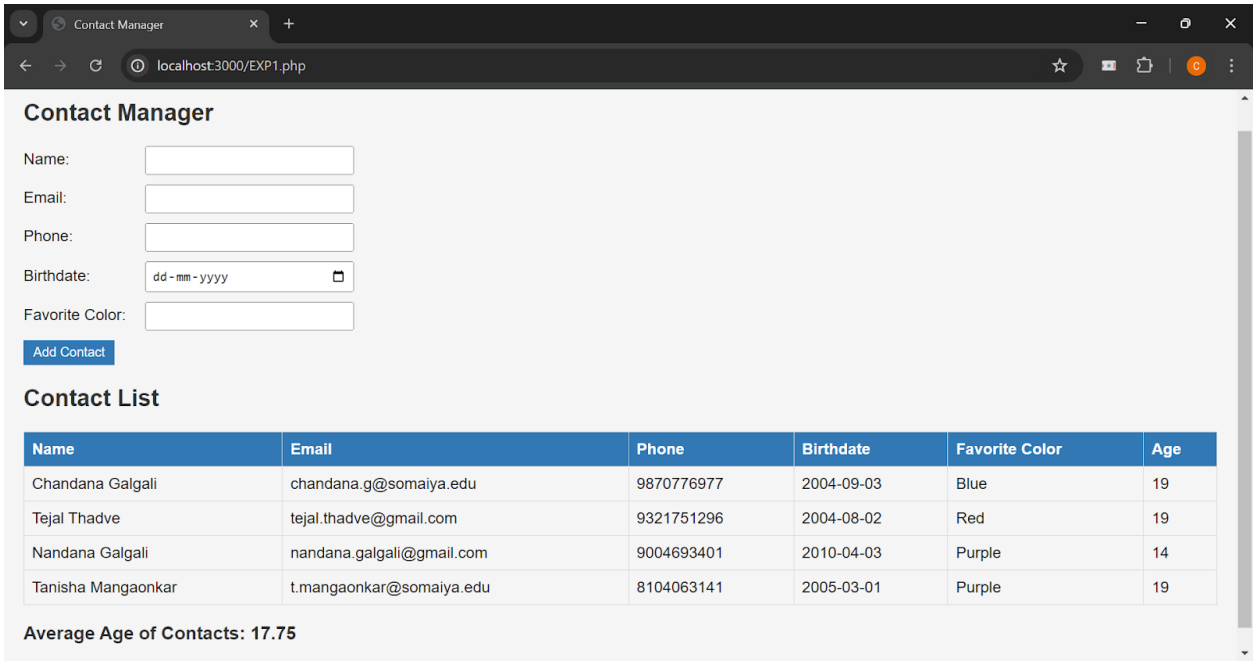
```
                    <th>Birthdate</th>
                    <th>Favorite Color</th>
                    <th>Age</th>
                    </tr>";
                $total_age = 0;
                $count = 0;
                while (($line = fgets($file)) !== false) {
                            list($name, $email, $phone, $birthdate, $color) =
explode(",", trim($line));
                    $age = calculate_age($birthdate);
                    $total_age += $age;
                    $count++;
                    echo "<tr>";
                    echo "<td>$name</td>";
                    echo "<td>$email</td>";
                    echo "<td>$phone</td>";
                    echo "<td>$birthdate</td>";
                    echo "<td>$color</td>";
                    echo "<td>$age</td>";
                    echo "</tr>";
                }
                echo "</table>";
                fclose($file);
                if ($count > 0) {
                    $average_age = $total_age / $count;
                      echo "<h3>Average Age of Contacts: " . round($average_age,
2) . "</h3>";
                }
            } else {
                echo "No contacts found.";
            }
        }

        display_contacts_and_average_age();
        ?>
    </div>
</body>
</html>
```

contacts.txt

```
Chandana Galgali,chandana.g@somaiya.edu,9870776977,2004-09-03,Blue
Tejal Thadve,tejal.thadve@gmail.com,9321751296,2004-08-02,Red
Nandana Galgali,nandana.galgali@gmail.com,9004693401,2010-04-03,Purple
Tanisha Mangaonkar,t.mangaonkar@somaiya.edu,8104063141,2005-03-01,Purple
```

Result:



Post Lab Questions:-

1. Explain Loops in PHP with examples.

Ans: Loops are used in programming to repeat a block of code as long as a specified condition is true. PHP supports four types of loops:

a. while Loop

The while loop executes a block of code as long as the specified condition is true.

Example:

```
$x = 1;
while($x <= 5) {
    echo "The number is: $x <br>";
    $x++;
}
```

This code will output the numbers from 1 to 5.

b. do...while Loop

The do...while loop will always execute the block of code once, then it will repeat the loop as long as the specified condition is true.

Example:

```
$x = 1;
do {
    echo "The number is: $x <br>";
    $x++;
} while ($x <= 5);
```

This code will also output the numbers from 1 to 5.

c. for Loop
The for loop is used when you know in advance how many times you want to execute a statement or a block of code.
Example:

```
for ($x = 0; $x <= 10; $x++) {
   echo "The number is: $x <br>";
}
```

This code will output the numbers from 0 to 10.

d. foreach Loop
The foreach loop works only on arrays and is used to loop through each key/value pair in an array.
Example:

```
$colors = array("red", "green", "blue", "yellow");
foreach ($colors as $value) {
   echo "$value <br>";
}
```

This code will output each color in the array.

2. Why choose PHP over other programming languages?
Ans: There are several reasons why PHP is chosen over other programming languages, especially for web development:

a. Ease of Use
PHP is relatively easy to learn compared to other programming languages. Its syntax is similar to C and Java, making it easier for developers with experience in those languages to pick up PHP quickly.

b. Wide Adoption and Community Support
PHP has been around for a long time and is widely used. This has resulted in a large community of developers who contribute to a vast repository of documentation, tutorials, and forums. This makes it easier to find solutions to common problems and get support when needed.

c. Compatibility
PHP is compatible with almost all servers used today (Apache, IIS, etc.) and works well with a variety of databases (MySQL, PostgreSQL, Oracle, etc.). This makes it a versatile choice for web development.
d. Cost-Effective
PHP is open-source and free to use. This significantly reduces the cost of development and deployment, especially for small businesses and individual developers.

e. Performance
PHP scripts execute on the server side. It is known for its speed and efficiency, especially when running on the LAMP stack (Linux, Apache, MySQL, PHP).

f. Frameworks and Tools
PHP has a variety of frameworks (like Laravel, Symfony, CodeIgniter) that streamline and enhance development processes. These frameworks provide a structured way to build web applications quickly and efficiently.

g. Flexibility
PHP allows developers to make changes in the code easily and quickly, making it ideal for dynamic websites where content changes frequently.

h. Integration Capabilities
PHP can easily be integrated with other technologies, including HTML, CSS, JavaScript, and various databases. It also supports integration with web services and APIs.

i. Security
While PHP has had some security issues in the past, the language and its frameworks have evolved to incorporate robust security features. Additionally, there are many best practices and tools available to secure PHP applications.

Outcomes: Illustrate use of basic PHP concepts to develop applications

---

Conclusion:-
The experiment aimed to demonstrate the use of basic PHP programming constructs such as data types, string functions, date and time functions, and math functions. The objectives were successfully achieved by implementing and executing the PHP code that showcased these constructs. The output matched the expected results, confirming the correct understanding and application of PHP basics.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

1. Thomson PHP and MySQL Web Development Addison-Wesley Professional , 5th Edition 2016.
2. Peter MacIntyre, Kevin Tatroe Programming PHP O'Reilly Media, Inc, 4th Edition 2020
3. Frank M. Kromann Beginning PHP and MySQL: From Novice to Professional, Apress 1st Edition, 2018