Name: Ansh Mehta

Roll No: 1814090

Batch: B1

WP-II: All Post lab Questions

POST LAB QUESTIONS

Exp 1: Simple PHP programs using Conditional Constructs

Q.1 Explain Loops in PHP with example.

Ans:

Often when you write code, you want the same block of code to run over and over again a certain number of times. So, instead of adding several almost equal code-lines in a script, we can use loops.

Loops are used to execute the same block of code again and again, as long as a certain condition is true.

In PHP, we have the following loop types:

- while - loops through a block of code as long as the specified condition is true
- do...while - loops through a block of code once, and then repeats the loop as long as the specified condition is true
- for - loops through a block of code a specified number of times
- foreach - loops through a block of code for each element in an array

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

Syntax:

```
while (condition is true) {
  code to be executed;
}
```

Example:

```
<?php
$x = 1;

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

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

Syntax

```
do {
  code to be executed;
} while (condition is true);
```

Example

```
<?php
$x = 1;

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

The for loop is used when you know in advance how many times the script should run.

Syntax

```
for (init counter; test counter; increment counter) {
  code to be executed for each iteration;
}
```

Parameters:

- *init counter*: Initialize the loop counter value
- *test counter*: Evaluated for each loop iteration. If it evaluates to TRUE, the loop continues. If it evaluates to FALSE, the loop ends.
- *increment counter*: Increases the loop counter value

Example

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

Q.2 Why choose PHP over other programming languages.

Ans:

We choose PHP over other programming languages because:

1. **1. PHP Is Simple and Fast**

PHP is known to be one of the simplest programming languages, any developer has ever come across! This is the major reason why most developers begin their career by mastering PHP.

The easy availability of the expert PHP developers through any PHP development company, is another fact, backing that PHP is easy to learn and gain expertise in.

Also, it is very easy for PHP to run fast as this language doesn’t use much of the system to run your web app. Therefore, the web applications of web pages load quite speedily.

Also, being one of the oldest programming languages, PHP has gotten rid of most of the glitches, making it one of the simplest, flawless, and fastest programming language for your web development or open source development solutions.

The simplicity of coding, easy learning, and a faster approach makes PHP a lot better than the other complex programming languages.

2. **2. Open Source Language With A Great Community Support**

PHP is an open source language, making web development an efficient and more interactive task for the developers. To get web development services in PHP, you can hire the best PHP web developer.

The vast open source network of PHP makes it the best programming language used by the experts to deliver cutting-edge open source development solutions.

The reason behind the increased efficiency and competence of PHP is the large community of developers, continuously making it better and better in terms of components, frameworks, and web development tools.

In case you have any issues during your web development process with PHP, you can get an instant solution through the extensive community available for this language.

The programming languages that are not open-source languages, don’t have a community support and the developer has to rely only on the vendor for help and support. However, that is not a case with PHP.

3. **3. Easy Integration**

PHP doesn’t work on any strict patterns or with any specific web apps or platforms. The web development with PHP is made much more extensive as the language integrates extremely well with multiple web apps. To name some, PHP integrates with the web applications like Pusher, Memcache, *MongoDB*, etc.

Such wide-ranging use and integration of PHP with different web apps, makes it one of the most widely used programming languages of the web development world. That is why, this powerful language fuels 30% of the entire web!

PHP is being used in almost every industry in one form or another. Any expert working in a PHP development company**,** can use the PHP language for web development in a plethora of industries like Government, Health, Banking, IT, Hospitality, Architecture, Large Corporations, and what not!

4. **4. Abundance of Frameworks**

No matter what kind of web development task you need to indulge into, you will always find the best-suited PHP framework.

Again, it all goes back to the fact that the PHP community is a huge and diverse community with the solution to every PHP problem. Because of such a vast community, there is a great availability of the PHP frameworks.

So, finding a PHP framework to help you with code reuse, session management, etc. is never a problem. You can get the help of the best PHP development company to find and use the most suitable PHP framework as per your web development needs. Some of the PHP frameworks are Zend, *Symfony*, Slim, Aiki, Phalcon, CakePHP, Yii, etc.

Exp 2: Use of Arrays and function in PHP

Q.3 What is difference between array_merge and array_combine?

Ans:

PHP have a vast collection of array functions which makes the use and manipulation of arrays much easier. This article defines the difference between two important array functions **array_merge** and **array_combine** which sounds similar but used for different operations.

array_merge() : array_merge() function merges the elements of one or more arrays into a single resultant array in such a way, So that the values of one array append to the end of previous ones. We can pass one or more arrays as parameters.

```
Syntax : array_merge(array1,array2...)
```

array1 : Specifies an input array. ***(Required)***
array2 : Specifies an input array. ***(Optional)***

Note :
1. If the arrays have same string keys, then the value of last array for that key will overwrite the value of previous ones.
2. If the arrays contain numeric keys, then the values of last array will not overwrite the values of previous arrays and will be append on last of the resultant array.
3. If there is only one array with numeric keys, Then the key of resultant array will be re-indexed and start from 0.

Example :

```
<?php

$array1 = array(1, 'a' => "red", 8, 'b' => "fruits");
$array2 = array('a' => "green", 2, 'xyz', 4, 'b' => "test");
$result = array_merge($array1, $array2);
print_r($result);

```

```
?>
```

Output :

```
Array ( [0] => 1 [a] => green [1] => 8 [b] => test [2] => 2 [3] => xyz [4] => 4 )
```

array_combine(): array_combine() function takes two arrays of same length and creates a new resultant array, using one array as keys and other array for values. The function returns the combined array on success and FALSE if the length of both array does not match.

```
Syntax : array_combine(array1,array2);
```

array1 : Specifies an input array. ***(Required)***
array2 : Specifies an input array. ***(Required)***

Note :
1. It is required that both arrays should have equal number of elements. If number of elements are not same on both arrays, the function returns FALSE.
2. The function issues E_WARNING and returns FALSE for empty arrays with PHP versions before 5.4.

Example :

```
<?php

$roll = array(1, 2, 3, 4);
$name = array("Amit", "Deepak", "Rahul", "Shyam");

$students = array_combine($roll,$name);
print_r($students);

?>
```

Output :

```
Array ( [1] => Amit [2] => Deepak [3] => Rahul [4] => Shyam )
```

Q.4 Write a program based on functions passing reference as parameter.

Ans:

By default, PHP variables are passed by value as the function arguments in PHP. When variables in PHP is passed by value, the scope of the variable defined at function level bound within the scope of function. Changing either of the variables doesn't have any effect on either of the variables.

Example:

```
<?php
// Function used for assigning new value to
// $string variable and printing it
function print_string( &$string ) {

	$string = "Function Ansh \n";

	// Print $string variable
	print( $string );
}

// Driver code
$string = "Global Ansh \n";
print_string( $string );
print( $string );
?>
```

Output:

```
Function Ansh
Global Ansh
```

Pass by reference:

When variables are passed by reference, use & (ampersand) symbol need to be added before variable argument. For example: function(&$x). Scope of both global and function variable becomes global as both variables are defined by same reference. Therefore, whenever global variable is change, variable inside function also gets changed and vice-versa is applicable.

Example:

```
<?php

function print_string( &$string ) {

	$string = "Function Ansh \n";

	print( $string );
}
```

```
// Driver code
$string = "Global Ansh \n";
print_string( $string );
print( $string );
?>
```

Output:

```
Function Ansh
Function Ansh
```

Exp 3: PHP programs on self-learning topics : String, Date, Time, Math,Image and File handling functions

Q.5 What is the purpose of php.ini file?

Ans:

At the time of PHP installation, **php.ini** is a special file provided as a default configuration file. It's very essential configuration file which controls, what a user can or cannot do with the website. Each time PHP is initialized, the **php.ini** file is read by the system. Sometimes you need to change the behavior of PHP at runtime, then this configuration file is to use.

All the settings related to register global variables, upload maximum size, display log errors, resource limits, the maximum time to execute a PHP script and others are written in a file as a set of directives which helps in declaring changes.

It helps in easy administration of web server using these configuration files. We can also write our own custom configuration files.

Q.6 How will you locate a string within a string in PHP?

Ans:

We are given two strings.We have to check whether the second string is a sub-string of the first string or not using PHP inbuilt function strpos().
The problem can be solved by iterating through the given string from 0 index to final length of the string and comparing the query string with the iterations. But in PHP we can also make use of some inbuilt functions to solve this particular problem.
strpos(): This function finds the position of the first occurrence of a string inside another string.
strlen() : Returns length of string.
If the index of first occurrence of the given string is within the length indices of the given string then the output returns true else the output returns false.
Example:

```
<?php
// PHP code to check if a string is
// substring of other
```

```
$s1 = "Bond, James Bond";
$s2 = "Bond";
if (strpos($s1, $s2) >= 0 &&
        strpos($s1, $s2) < strlen($s1))
        echo("True");
else
        echo("False");
?>
```

Output:

True

Q.7 How can you display a file download dialog box using PHP?

Ans:

In order to display a download dialog for pdf file rather than opening it in the browser, we can put the following snippet of code in a php file and name the file download.php.

The path to the pdf file is specified in $filename variable. You can also pass filename as a parameter in the URL but you will need to check for Cross Site Scripting (XSS) and various script injection attempts if you decide to get the filename from the URL paramater.

```
$filename = '/path/to/your/file/download.pdf';
header("Pragma: public");
header("Expires: 0");
header("Pragma: no-cache");
header("Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0");
header("Content-Type: application/force-download");
header("Content-Type: application/octet-stream");
header("Content-Type: application/download");
header('Content-disposition: attachment; filename=' . basename($filename));
header("Content-Type: application/pdf");
header("Content-Transfer-Encoding: binary");
header('Content-Length: ' . filesize($filename));
@readfile($filename);
exit(0);
```

Exp 4: Form Handling using PHP

Q.8 Compare and Contrast GET and POST method.

Ans:

The following table compares the two HTTP methods: GET and POST.

	GET	**POST**
BACK button/Reload	Harmless	Data will be re-submitted (the browser should alert the user that the data are about to be re-submitted)
Bookmarked	Can be bookmarked	Cannot be bookmarked
Cached	Can be cached	Not cached
Encoding type	application/x-www-form-urlencoded	application/x-www-form-urlencoded or multipart/form-data. Use multipart encoding for binary data
History	Parameters remain in browser history	Parameters are not saved in browser history
Restrictions on data length	Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters)	No restrictions
Restrictions on data type	Only ASCII characters allowed	No restrictions. Binary data is also allowed

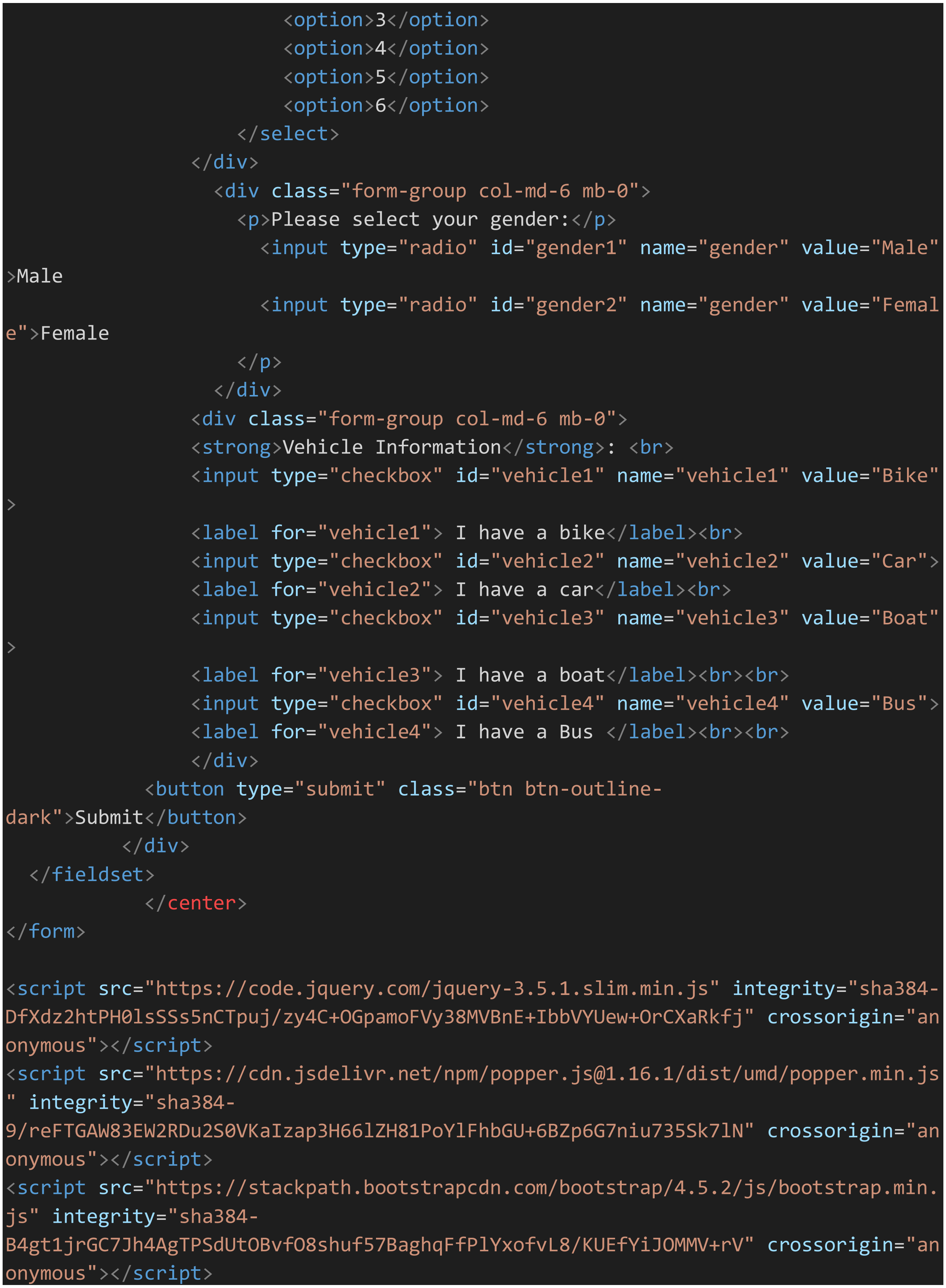
Security	GET is less secure compared to POST because data sent is part of the URL Never use GET when sending passwords or other sensitive information!	POST is a little safer than GET because the parameters are not stored in browser history or in web server logs
Visibility	Data is visible to everyone in the URL	Data is not displayed in the URL

Q.9 Write a script to handle checkbox (Multiple Values) and drop down Form Inputs.

Ans:

HMTL Code:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css" integrity="sha384-JcKb8q3iqJ61gNV9KGb8thSsNjpSL0n8PARn9HuZOnIxN0hoP+VmmDGMN5t9UJ0Z" crossorigin="anonymous">
    <title>
        Home
    </title>
</head>
<body  style="background-color:white;">
<form  method="POST" action="Forms.php">
    <center style="padding-top:100px;">
    <fieldset style="background: rgba(0,0,0,0.35);padding: 2vw;border-radius: 1vw; width:65%;">
            <label for="FULLNAME">Full Name:</label>
            <input type="text" name="FULLNAME" id="FULLNAME"></input>
                  <div class="form-group col-md-6 mb-0">
                    <label for="FName">Enter the number whose Factorial is to be found:</label>
                    <select name="FName" id="FName">
                        <option>2</option>
```



```
                        <option>3</option>
                        <option>4</option>
                        <option>5</option>
                        <option>6</option>
                    </select>
                </div>
                  <div class="form-group col-md-6 mb-0">
                    <p>Please select your gender:</p>
                      <input type="radio" id="gender1" name="gender" value="Male"
>Male
                      <input type="radio" id="gender2" name="gender" value="Femal
e">Female
                    </p>
                  </div>
                <div class="form-group col-md-6 mb-0">
                <strong>Vehicle Information</strong>: <br>
                <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike"
>
                <label for="vehicle1"> I have a bike</label><br>
                <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
                <label for="vehicle2"> I have a car</label><br>
                <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat"
>
                <label for="vehicle3"> I have a boat</label><br><br>
                <input type="checkbox" id="vehicle4" name="vehicle4" value="Bus">
                <label for="vehicle4"> I have a Bus </label><br><br>
                </div>
            <button type="submit" class="btn btn-outline-
dark">Submit</button>
          </div>
  </fieldset>
            </center>
</form>

<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js" integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj" crossorigin="an
onymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.1/dist/umd/popper.min.js
" integrity="sha384-
9/reFTGAW83EW2RDu2S0VKaIzap3H66lZH81PoYlFhbGU+6BZp6G7niu735Sk7lN" crossorigin="an
onymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.
js" integrity="sha384-
B4gt1jrGC7Jh4AgTPSdUtOBvfO8shuf57BaghqFfPlYxofvL8/KUEfYiJOMMV+rV" crossorigin="an
onymous"></script>
```

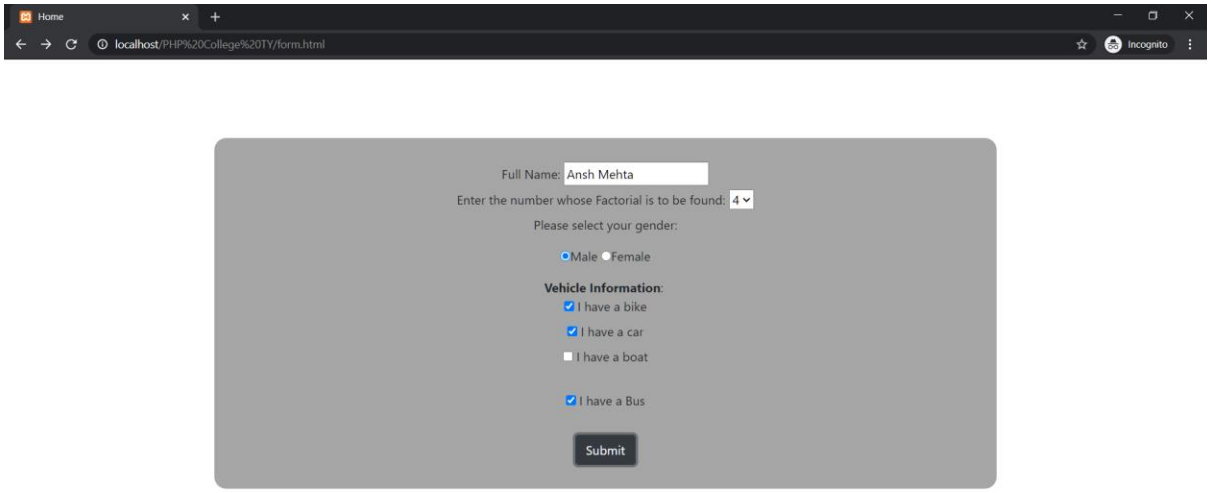
```
</body>
</html>
```

PHP Code:

```
<?php
$xs=$_POST['FULLNAME'];
echo $xs."<br><br>";

$x1=$_POST["FName"];
echo $x1;
$fact=1;
while($x1 >= 1) {
    $fact=$fact*$x1;
    $x1=$x1-1;
}
$x1=$_POST["FName"];
echo "<br>Factorial of ".$x1." is: ";
echo $fact."<br><br>";

$male_status = 'unchecked';
$female_status = 'unchecked';


$selected_radio = $_POST['gender'];

if ($selected_radio == 'Male')
{

echo "Male";

}
else if ($selected_radio == 'Female') {

echo "Female";

}
echo "<br><br>";
$VO=0;

if (isset($_POST["vehicle1"]))
{
    echo "You own a Bike!!<br>";
    $VO=1;
}

if (isset($_POST["vehicle2"]))
```

```
{
    echo "You own a Car!! <br>";
    $VO=1;
}
if (isset($_POST["vehicle3"]))
{
    echo "You own a Boat!!<br>";
    $VO=1;
}
if (isset($_POST["vehicle4"]))
{
    echo "You own a Bus!!<br>";
    $VO=1;
}

if ($VO==0)
{
    echo "You don't own any vehicle";
}
?>
```

Output:



Full Name: Ansh Mehta

Enter the number whose Factorial is to be found: 4

Please select your gender:

● Male ○ Female

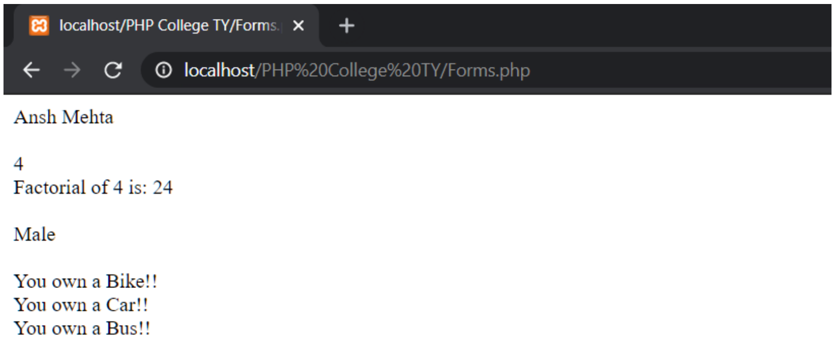
Vehicle Information:

☑ I have a bike

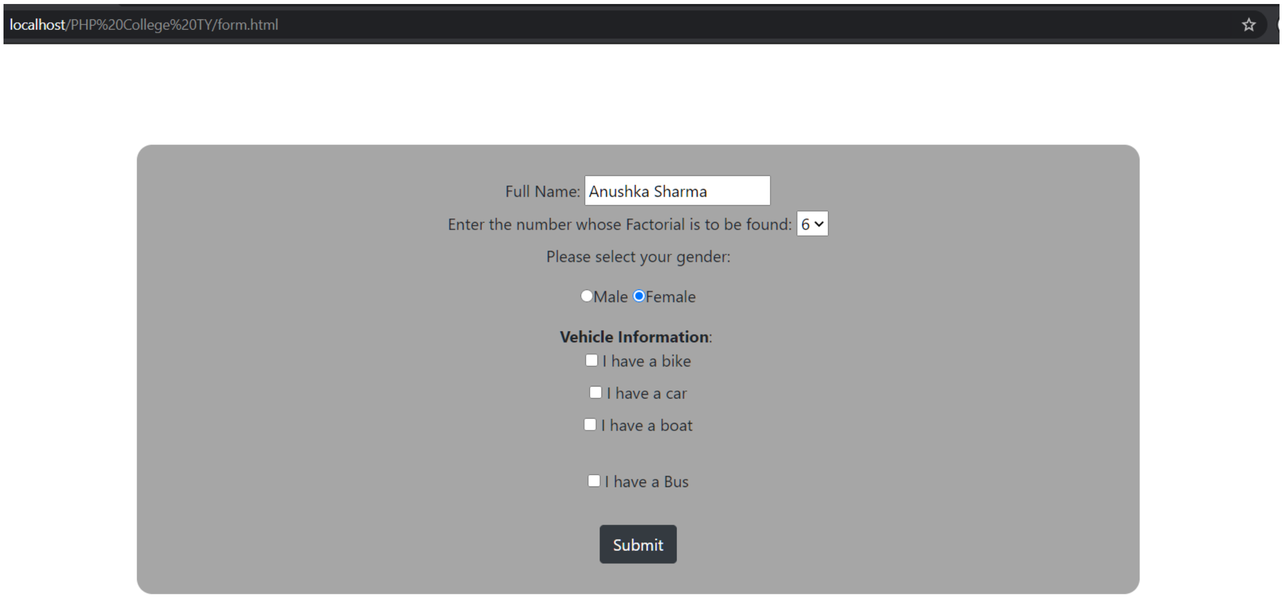
☑ I have a car

□ I have a boat

☑ I have a Bus


localhost/PHP College TY/Forms.
localhost/PHP%20College%20TY/Forms.php

```
Ansh Mehta

4
Factorial of 4 is: 24

Male

You own a Bike!!
You own a Car!!
You own a Bus!!
```



Full Name: Anushka Sharma

Enter the number whose Factorial is to be found: 6

Please select your gender:

○ Male ● Female

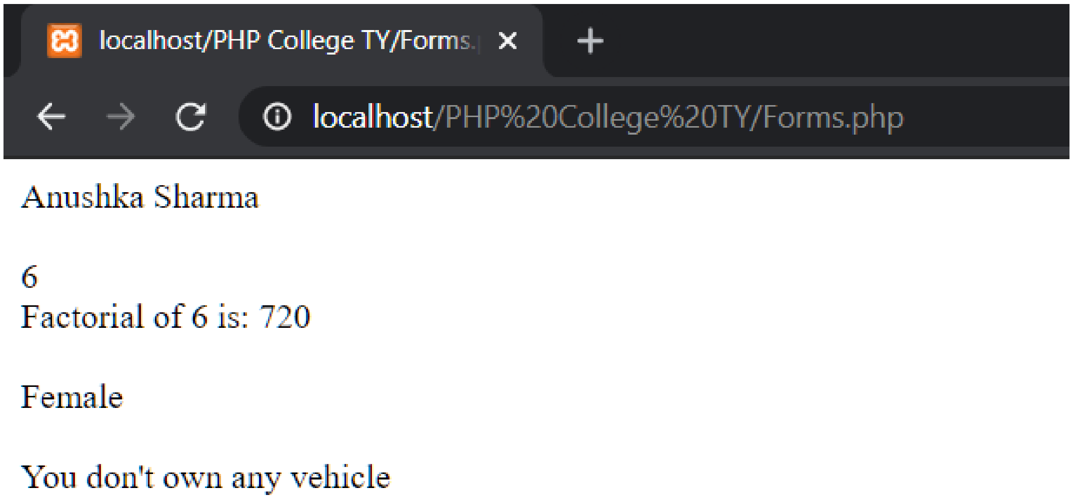
Vehicle Information:

□ I have a bike

□ I have a car

□ I have a boat

□ I have a Bus


localhost/PHP College TY/Forms.
localhost/PHP%20College%20TY/Forms.php

```
Anushka Sharma

6
Factorial of 6 is: 720

Female

You don't own any vehicle
```

Exp 5: Form Validation using PHP

Q.10 What the significance of $_SERVER["PHP_SELF"] and htmlspecialchars() function?

Ans)

$_SERVER['PHP_SELF']:

The $_SERVER["PHP_SELF"] is a super global variable that returns the filename of the currently executing script. the $_SERVER["PHP_SELF"] sends the submitted form data to the page itself, instead of jumping to a different page. This way, the user will get error messages on the same page as the form.

Htmlspecialchars():

The htmlspecialchars() function converts special characters to HTML entities. This means that it will replace HTML characters like < and > with &lt; and &gt;. This prevents attackers from exploiting the code by injecting HTML or Javascript code (Cross-site Scripting attacks) in forms.

Q. 11 Explain working of empty() and isset() with an example. Compare the working of both these functions

Ans) *isset* — Determine if a variable is set and is not NULL

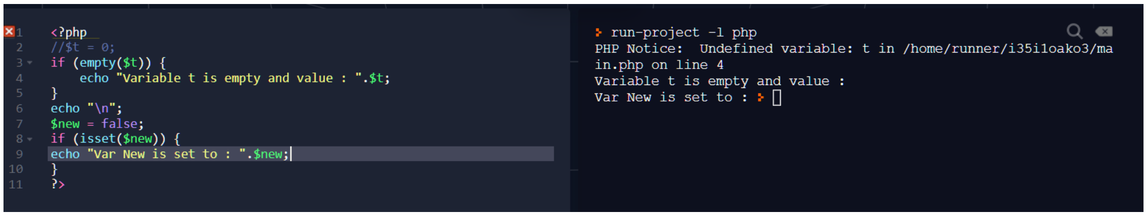
The isset() function is an inbuilt function in PHP which checks whether a variable is set and is not NULL. This function also checks if a declared variable, array or array key has null value, if it does, isset() returns false, it returns true in all other possible cases.

Syntax: bool isset($var, mixed)

Parameters: This function accepts more than one parameters. The first parameter of this function is $var.

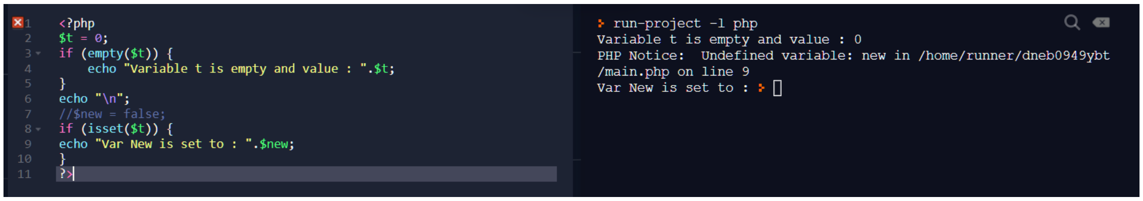
empty — Determine whether a variable is empty

The empty() function is a language construct to determine whether the given variable is empty or NULL. The !empty() function is the negation or complement of empty() function. The empty() function is considerably equal to !isset() function and !empty() function is equal to isset() function.



```php
<?php
//$t = 0;
if (empty($t)) {
    echo "Variable t is empty and value : ".$t;
}
echo "\n";
$new = false;
if (isset($new)) {
echo "Var New is set to : ".$new;
}
?>
```

```
› run-project -l php
PHP Notice:  Undefined variable: t in /home/runner/i35i1oako3/ma
in.php on line 4
Variable t is empty and value :
Var New is set to : › 
```



```php
<?php
$t = 0;
if (empty($t)) {
    echo "Variable t is empty and value : ".$t;
}
echo "\n";
//$new = false;
if (isset($t)) {
echo "Var New is set to : ".$new;
}
?>
```

```
› run-project -l php
Variable t is empty and value : 0
PHP Notice:  Undefined variable: new in /home/runner/dneb0949ybt
/main.php on line 9
Var New is set to : › 
```

Why use both:

The isset() and !empty() functions are similar and both will return the same results. But the only difference is !empty() function will not generate any warning or e-notice when the variable does not exists. It is enough to use either of the function. By incorporating both functions in a program causes time lapse and unnecessary memory usage.

Exp 6: Cookies and Session handling using PHP

Q.12 Differentiate between cookies and sessions in PHP

Cookie	Session
• Cookie ends depending on the lifetime you set for it	• A session ends when a user closes his browser
• You don't need to start cookie as it is stored in your local machine	• In PHP, before using $_SESSION, you have to write session_start(); Likewise for other languages
• A cookie is not dependent on Session	• A session is dependent on Cookie
• The official maximum cookie size is 4KB	• Within-session you can store as much data as you like. The only limits you can reach is the maximum memory a script can consume at one time, which is 128MB by default
• Cookies are client-side files that contain user information	• Sessions are server-side files which contain user information

Q.13 Explain the process of destroying a single session variable and the process of destroying the entire session with suitable examples of each.

A PHP session can be destroyed by session_destroy() function. This function does not need any argument and a single call can destroy all the session variables.

Syntax: bool session_destroy(void)

Eg: session.php

```
<?php
echo "DESTROY WHOLE SESSION ";
echo "<br>";
session_start();
$_SESSION['counter1']="one";
$_SESSION['counter2']="two";

echo "BEFORE SESSION DESTROY FUNCTION";
echo "<br>";
echo $_SESSION["counter1"];
echo "<br>";
echo $_SESSION["counter2"];
?>
```

Session1.php

```
<?php
echo "<br>";
session_destroy();
echo "AFTER SESSION DESTROY FUNCTION";

echo $_SESSION['counter1'];

echo $_SESSION['counter2'];
?>
```

If you want to destroy a single session variable then you can use unset() function to unset a session variable.

Syntax: bool session_unset(void)

Eg:

```
<?php
echo "DESTROY SINGLE SESSION VARIABLE";
echo "<br>";
session_start();
$_SESSION['counter']=1;
echo "BEFORE SESSION DESTROY VARIABLE";
echo "<br>";
echo $_SESSION['counter'];
echo "<br>";
echo "AFTER UNSET SESSION DESTROY VARIABLE";
```

```
unset($_SESSION['counter']);
echo $_SESSION['counter'];
?>
```

Exp 7: Database connectivity in PHP

Q. 14 What is other alternative to connect to any of the database using PHP?

Ans)

One can connect to database using:

- MySQLi extension
- PDO (PHP Data Objects)

Connection using MySQLi Extension

```
<?php
$servername = "localhost";
$username = "root";
$password = " ";

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

// Check connection
if ($conn->connect_error) {
  die("Connection failed: " . $conn->connect_error);
}
echo "Connected successfully";
?>
```

Connection using PDO

```
<?php
$servername = "localhost";
$username = "root";
$password = "";
```

```
try {
  $conn = new PDO("mysql:host=$servername;dbname=myDB", $username, $password);
  // set the PDO error mode to exception
  $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
  echo "Connected successfully";
} catch(PDOException $e) {
  echo "Connection failed: " . $e->getMessage();
}
?>
```

Q. 15 How does PHP supports Postgresql connectivity??

Ans)

The PostgreSQL functions in the previous procedure can only be used with PostgreSQL databases. PDO abstracts database access, and enables you to use code that can handle different types of databases. To connect to PostgreSQL using PDO, follow these steps:

1. Use the following PHP code to connect to PostgreSQL and select a database. Replace username with your username, password with your password, and dbname with the database name:

```
<?php

 $myPDO = new PDO('pgsql:host=localhost;dbname=dbname', 'username', '
password');

?>
```

2. After the code connects to PostgreSQL and selects the database, you can run SQL queries and perform other operations. For example, the following PHP code runs a SQL query that extracts the last names from the *employees* table, and stores the result in the *$result* variable:

```
<?php

$result = $myPDO->query("SELECT * FROM employees");

?>
```

Exp 7: REST APIs IN PHP

Q.16 What is the purpose of REST API?

Ans)

A RESTful API is an architectural style for an application program interface (API) that uses HTTP requests to access and use data. That data can be used to GET, PUT, POST and DELETE data types, which refers to the reading, updating, creating and deleting of operations concerning resources. An API for a website is code that allows two software programs to communicate with each other. The API spells out the proper way for a developer to write a program requesting services from an operating system or other application. A RESTful API -- also referred to as a RESTful web service or REST API -- is based on representational state transfer (REST), which is an architectural style and approach to communications often used in web services development. REST technology is generally preferred over other similar technologies. This tends to be the case because REST uses less bandwidth, making it more suitable for efficient internet usage. RESTful APIs can also be built with programming languages such as JavaScript or Python. The REST used by browsers can be thought of as the language of the internet. With cloud use on the rise, APIs are being used by cloud consumers to expose and organize access to web services. REST is a logical choice for building APIs that allow users to connect to, manage and interact with cloud services flexibly in a distributed environment. RESTful APIs are used by such sites as Amazon, Google, LinkedIn and Twitter.

Q.17 What are the other alternatives to REST API?

Ans) There are many alternatives for REST APIs like FALCOR, gRPC, Apache thrift but *the most preferred alternative is graphQL.* GraphQL is preferred as client has the ability to dictate what exactly they need as an output. Hence it is preferred.

- **REST API:** It is an Application Program Interface (API) that uses HTTP protocols to perform operations on data.
- **HTTP API:** Any API that uses HTTP protocols for manipulation of data are called as HTTP APIs.
- **GraphQL API:** GraphQL is a query language for APIs and a runtime for fulfilling those queries with your existing data. APIs developed using GraphQL are called GraphQL APIs.