**Name: Aarya Tiwari**

**Batch: B2**

**Roll no.: 16010421119**

**Experiment No. : 2**

Code:

1. Indexed Array

<?php

    $frands = array("Aarya","Dhairya","Gonka","Dhrumvv");

    echo $frands[0] . "<br><br>";

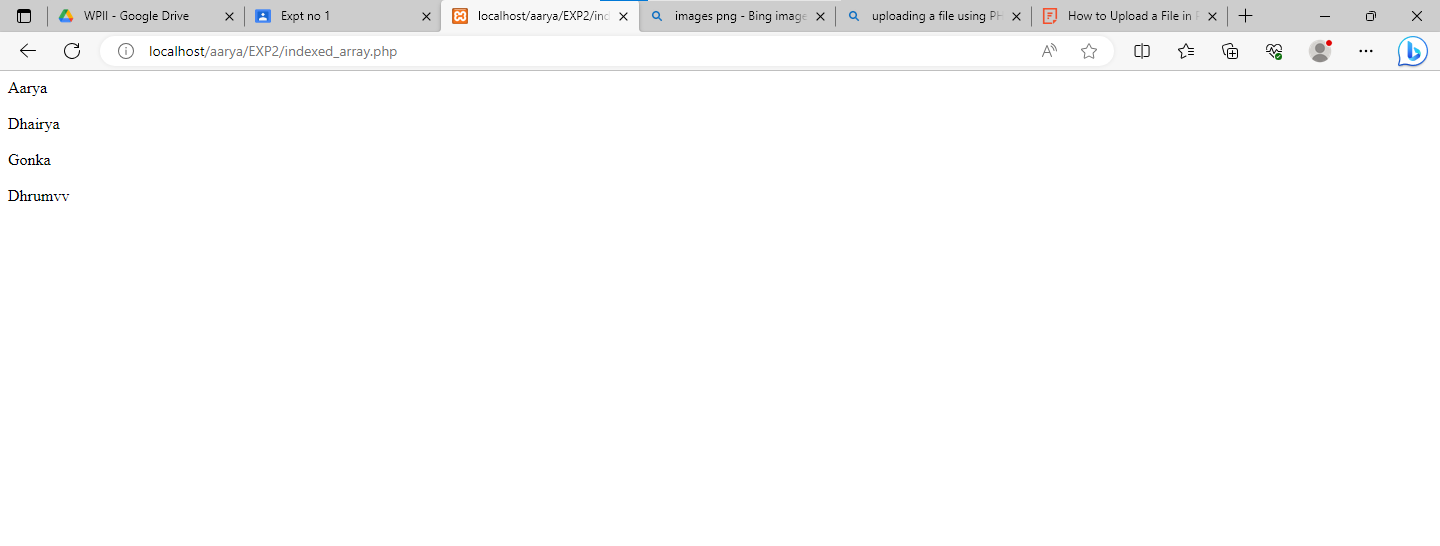
    echo $frands[1] . "<br><br>";

    echo $frands[2] . "<br><br>";

    echo $frands[3] . "<br><br>";

?>

Output:



1. Associative Array

<?php

    $marks = array("Aarya"=>9.5,"Dhairya"=>9.5,"Gonka"=>8.5,"Dhrumvv"=>8.5);

    echo "Marks for Aarya is " . $marks["Aarya"] . "<br><br>";

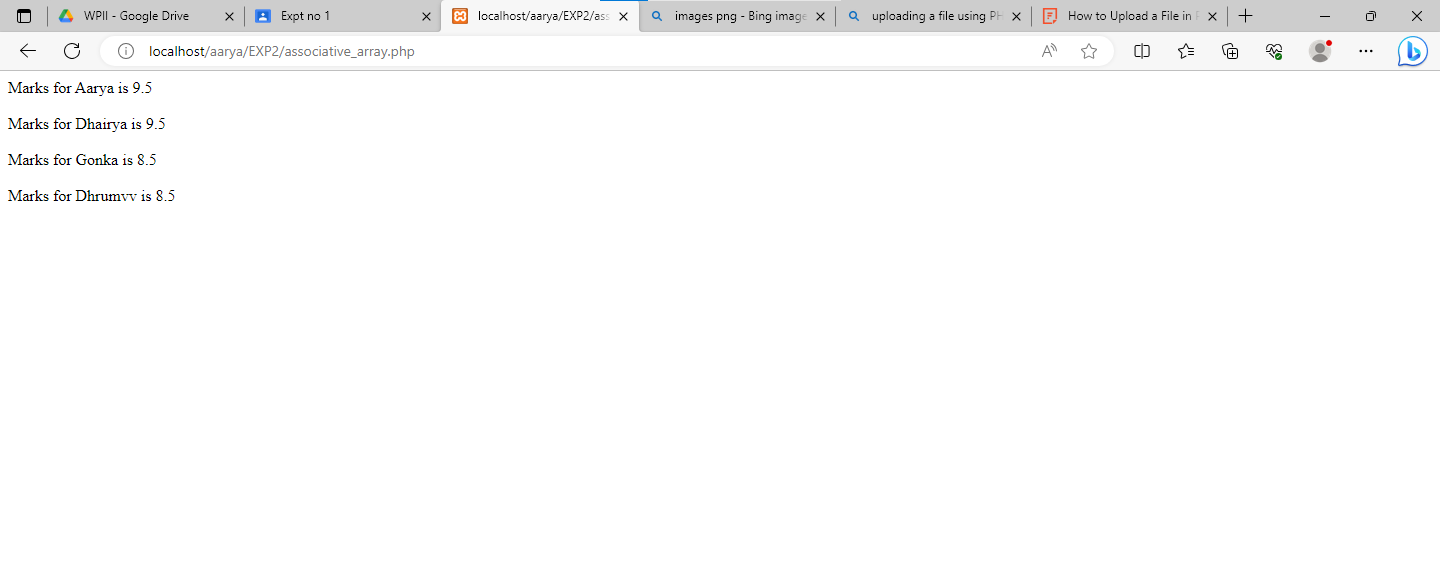
    echo "Marks for Dhairya is " . $marks["Dhairya"] . "<br><br>";

    echo "Marks for Gonka is " . $marks["Gonka"]. "<br><br>";

    echo "Marks for Dhrumvv is " . $marks["Dhrumvv"]. "<br><br>";

?>

Output:



1. Multidimensional Array

<?php

    $employee = array(

        array(1,"Aarya","12/11/2003"),

        array(2,"Dhairya","31/10/2003"),

        array(3,"Gonka","18/03/2003"),

        array(4,"Dhrumvv","13/07/2003"),

    );

    for($row = 0;$row<4;$row++)

    {

        echo "Details of Employee ".$row+1 ." are as follows:- <br>";

        echo "Id of the employee: " . $employee[$row][0] . "<br>";

        echo "Name of the employee: " . $employee[$row][1] . "<br>";

        echo "Date of Birth of the employee: " . $employee[$row][2] . "<br>";

        echo "<br><br>";

    }

    // Tabular Form

    for($row = 0;$row<4;$row++){

        for($col = 0;$col<3;$col++)

        {

            echo $employee[$row][$col] . " ";

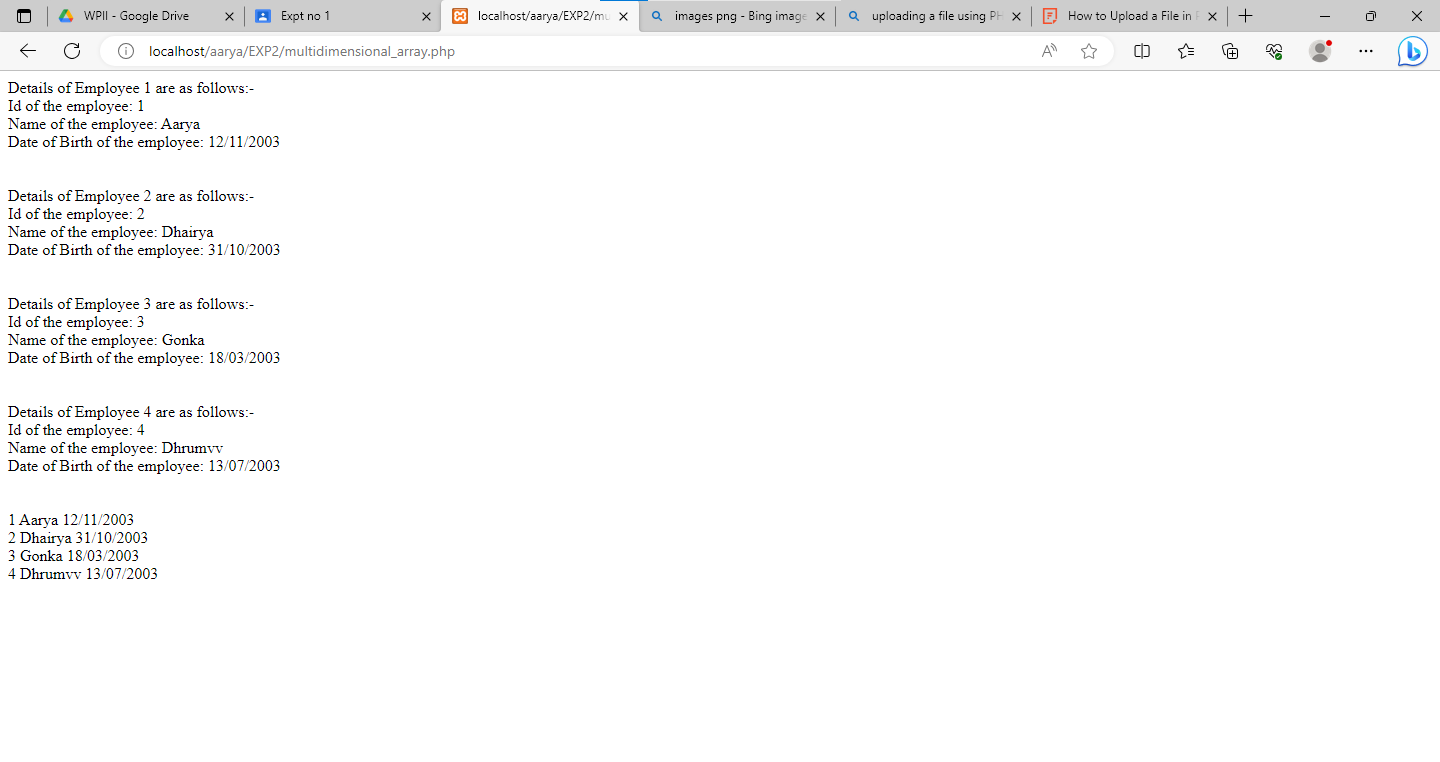
        }

        echo "<br>";

    }

?>

Output:



1. Array Functions

<?php

    function printArray($arr)

    {

        echo "<br>";

        foreach($arr as $a)

        {

            echo "$a <br>";

        }

    }

    $b2 = array("Aarya","Dev","Smit","Mihir","Siddarth","Riya","Hussain","Advait","Parth","Sumit","Om","Tanmay","Prathamesh","Arpan","Twisha");

    $b2frands = array("Aarya","Dhairya","Smit","Gonka","Dhrumvv");

    echo "The people in B2 are: ";

    printArray($b2);

    //Changing Key Cases

    // echo "After changing the key cases";

    // $b2case = array\_change\_key\_case($b2,CASE\_UPPER);

    // printArray($b2case);

    // echo "<br><br>";

    //Array Chunk Function

    print\_r(array\_chunk($b2,3));

    echo "<br><br>";

    //Count Function

    echo "The Number of elements in the Array is: " . count($b2) . "<br>";

    echo "<br><br>";

    //Sort Function

    sort($b2);

    echo "The Sorted Array is: ";

    printArray($b2);

    // print\_r($b2);

    echo "<br><br>";

    //Reversing the Array

    $reverseArray = array\_reverse($b2);

    echo "The Reversed Array is: ";

    printArray($reverseArray);

    // print\_r($reverseArray);

    echo "<br><br>";

    //Searching in An Array

    $key = array\_search("Dev",$b2);

    // $key+=1;

    gettype($key);

    if($key == false)

    {

        echo "Dev was not found in the Array <br><br>";

    }

    else

    {

        echo "Dev is at position $key in the Array";

        echo "<br><br>";

    }

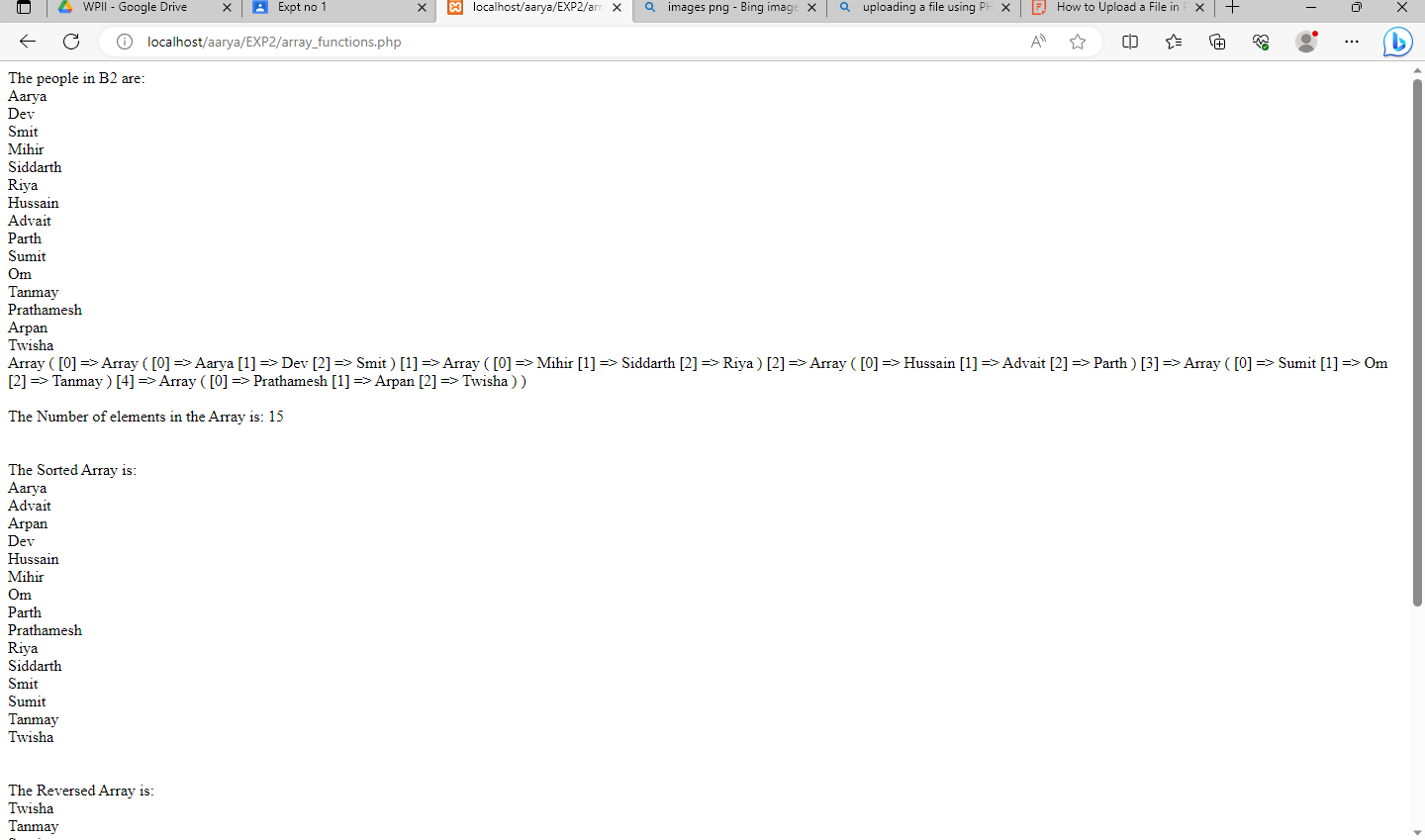
    //Intersection of an Array

    $prevb2 = array\_intersect($b2,$b2frands);

    echo "The previous members in the B2 Batch are: ";

    print\_r($prevb2);

?>



1. User-Defined Functions

    function printArray($arr)

    {

        echo "<br>";

        foreach($arr as $a)

        {

            echo "$a <br>";

        }

    }

1. File handling

<?php

    $file1 = fopen("index1.txt",'r+');

    $file2 = fopen("index2.txt",'r+');

    $file3 = fopen("index3.txt",'r+');

    $filesize1 = filesize("index1.txt");

    $filesize2 = filesize("index2.txt");

    $filesize3 = filesize("index3.txt");

    $filedata1 = fread($file1, $filesize1);

    $filedata2 = fread($file2, $filesize2);

    $filedata3 = fread($file3, $filesize3);

*//Reading from a file..*

    echo $filedata1 . "<br>";

    echo $filedata2 . "<br>";

    echo $filedata3 . "<br>";

    $file = fopen("index1.txt", 'r+');

    $text = "Dev Dev Dev Dev Dev Dev Dev Dev\n";

    fwrite($file, $text);

    $filesize = filesize("index1.txt");

    $filedata = fread($file, $filesize1);

    echo $filedata;

*//Creating/Writing in a file..*

    $myfile = fopen("newfile.txt", "w") or die("Unable to open file!");

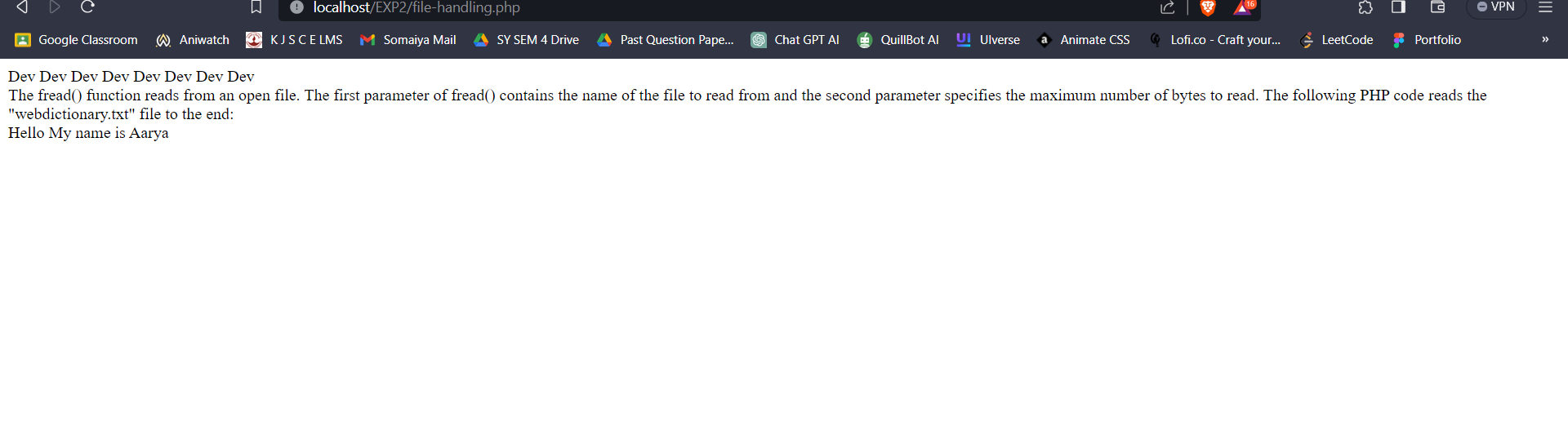
    $txt = "I want to write this text into the file\n";

    fwrite($myfile, $txt);

    fclose($myfile);

?>

Output:



1. Image Handling

<?php

    header("Content-Type: image/png");

    $im = @imagecreate(500,500)

    or die("Cannot Initialize new GD image stream");

    $background\_color = imagecolorallocate($im, 255, 255, 255);

    $text\_color = imagecolorallocate($im, 14, 233, 91);

    imagestring($im, 5, 200, 200, "Dev Dev Dev Dev Dev", $text\_color);

    imagepng($im);

    imagedestroy($im);

?>

Output:



1. Image Upload

<?php

    if(isset($\_FILES['image']))

    {

        $errors= array();

        $file\_name = $\_FILES['image']['name'];

        $file\_size =$\_FILES['image']['size'];

        $file\_tmp =$\_FILES['image']['tmp\_name'];

        $file\_type=$\_FILES['image']['type'];

        $file\_ext=strtolower(end(explode('.',$\_FILES['image']['name'])));

        $extensions= array("jpeg","jpg","png","jfif");

        if(in\_array($file\_ext,$extensions)=== false)

        {

            $errors[]="This extension is not supported by our system, please choose a JPEG or PNG file.";

        }

        if($file\_size > 2097152)

        {

            $errors[]='File size must be exactly 2 MB';

        }

        if(empty($errors)==true)

        {

            move\_uploaded\_file($file\_tmp,"images/".$file\_name);

            echo "Success";

        }

        else

        {

            print\_r($errors);

        }

    }

?>

<html>

<body>

<form action="" method="POST" enctype="multipart/form-data">

<input type="file" name="image" />

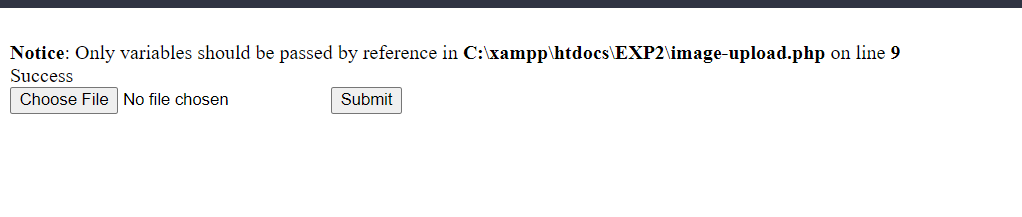
<input type="submit"/>

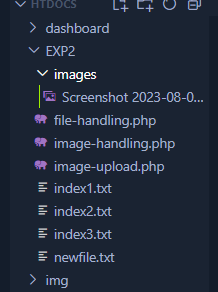
</form>

</body>

</html>

Output:





**Post Lab Questions:**

**1. What is difference between array\_merge and array\_combine?**

Ans: array\_merge and array\_combine are both functions in PHP used for working with arrays, but they serve different purposes and have different behaviors:

array\_merge():

array\_merge is used to merge two or more arrays into a single array.

It takes multiple arrays as arguments and combines their values into a single array.

Duplicate values are preserved, and numeric keys are re-indexed starting from zero.

array\_combine():

array\_combine is used to create a new array by using one array for keys and another array for values.

It takes two arrays as arguments, one for keys and one for values.

The number of elements in both arrays should be the same; otherwise, it will result in an error.

**2. Write a program based on functions passing reference as parameter.**

**Ans:**

**Code:**

<?php

    function square(&$num)

    {

        $num = $num \* $num;

    }

    $value = 5;

    echo "Before: $value\n"; *// Before: 5*

    square($value);

    echo "After: $value\n"; *// After: 25*

?>

**3. How can you display a file download dialog box using PHP?**

**Ans:**

**Code:**

<?php

    $file\_path = 'path/to/your/file.pdf'; *// Replace with the actual path to your file*

    $file\_name = 'downloaded\_file.pdf'; *// The name you want the downloaded file to have*

*// Check if the file exists*

    if (file\_exists($file\_path)) {

*// Set headers to trigger the download dialog*

        header('Content-Type: application/octet-stream');

        header('Content-Disposition: attachment; filename="' . $file\_name . '"');

        header('Content-Length: ' . filesize($file\_path));

*// Read and output the file content*

        readfile($file\_path);

        exit;

    } else {

*// File not found error handling*

        echo "File not found.";

    }

?>

**4. What is the purpose of php.ini file?**

Ans: The php.ini file is a configuration file used in PHP to control various aspects of the PHP runtime environment. It plays a crucial role in customizing the behavior of PHP for your web application. The name "php.ini" stands for "PHP initialization."

The main purposes of the php.ini file are:

1. Configuration Settings: The php.ini file contains a wide range of configuration settings that determine how PHP behaves. These settings include things like memory limits, execution timeouts, error reporting levels, database connections, file upload limits, and more. By modifying these settings, you can fine-tune PHP to meet the specific requirements of your application.
2. Server-Wide Configuration: The php.ini file is used to configure PHP at a server-wide level. This means that changes made in this file affect all PHP scripts running on the server. It's especially useful when you want to enforce certain settings consistently across multiple applications.
3. Security: Many security-related settings are controlled through the php.ini file. For example, you can disable certain functions that might pose security risks, enable/disable certain security modules like the Suhosin extension, and control how PHP handles potentially harmful input data.
4. Performance Optimization: The php.ini file allows you to configure PHP for optimal performance. You can adjust settings related to caching, opcode optimization, and memory management to ensure that your PHP scripts run efficiently.

**Outcomes:**

CO1: Illustrate use of basic PHP concepts to develop applications

**Conclusion:**

We can conclude that we have learnt about Image Handling , File Handling , User-defined functions and Types of Arrays.

**References:**

https://www.w3schools.com/php/php\_file\_create.asp