**VIVEKANANDA INSTITUTE OF PROFESSIONAL STUDIES**

**VIVEKANANDA SCHOOL OF INFORMATION TECHNOLOGY**



**BACHELOR OF COMPUTER APPLICATION**

**Web Based Programming Lab File**

**Paper Code: BCA 172**



**SUBMITTED TO: SUBMITTED BY:**

Dr. Anupama Jha Name: Himanshu Chandna

Assistant Professor Enrollment no:06517702023

VSIT, VIPS BCA-II (B)

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Name of the Program** | **Date** | **Signature** |
| 1 | Assignment - 1 | 30 January 2024 |  |
| 2 | Assignment – 2 | 6 February 2024 |  |
| 3 | Assignment – 3 | 13 February 2024 |  |
| 4 | Assignment – 4 | 27 February 2024 |  |
| 5 | Assignment – 5 | 5 March 2024 |  |
| 6 | Assignment – 6 | 5 March 2024 |  |
| 7 | Assignment – 7 | 5 March 2024 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**INDEX**

**Final List of Web Based Programming PHP Lab Assignment Paper code: BCA 172**

**Faculty Name: Dr. Anupama Jha**

**(Jan-May 2024)**

1. WAP to show the usage of nested if statement to find the maximum of 3 numbers.

2. WAP to show the usage of switch-case and for/while/do while loop to create a menu driven program for calculator.

3. WAP to implement all the inbuilt array functions like array\_pad(), array\_slice(), array\_splice(), list() functions. (use foreach wherever applicable)

4. Write a program to perform all four types of sorting.

5. WAP using PHP to implement regular expressions including modifiers, operators and metacharacters.

6. Design a login form and validate using PHP programming. (Use of regular expression, Super global variables such as $\_GET(), $\_POST(), $\_REQUEST() andappropriate functions)

7. Design a login form and validate using PHP programming (use of PHP Filters).

8. WAP that passes the control to another page (include, require, exit and die function).

9. Explain Cookie and how to make login logout PHP script using Cookie. The following first screenshot is for file named ***login.php***. Do necessary validation on form inputs: user’s email and password.

A screenshot of a login page

Description automatically generated

The screenshot for ***second.php*** is as per the format after successful logged into to thepage. Implement Logout option with cookie.

A white background with black text

Description automatically generated

10. Explain Session and write a PHP script to count the number of visits of web  pageby the user session.

11. Write a menu-driven program for:Choice a) Create a file.

Choice b) To read such file and then copy the content of this file to another file. Choice c) To read and display the content of the previously created file. Choice d) To modify the content of a file.

12. Design a form that upload, download and display image file using PHP script. 13. WAP to implement OOPS Concept.

14. Write the following scripts using PHP and MySQL:

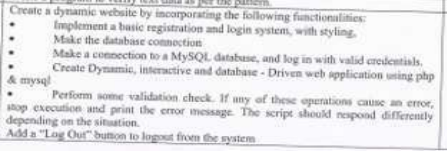
i) To create a MySQL Database

ii) To create a table and insert few records into it using forms

iii) To select all the records and display it in table.

iv) To modify a table (add/delete/modify columns)

15.



1. WAP to show the usage of nested if statement to find the maximum of 3 numbers.

Source Code:

<?php

// Allowing the user to enter three numbers

$num1 = readline("Enter first number: ");

$num2 = readline("Enter second number: ");

$num3 = readline("Enter third number: ");

// Finding the greatest number using nested if statements

if ($num1 >= $num2) {

    if ($num1 >= $num3) {

        echo "The greatest number is: " . $num1 . "\n";

    } else {

        echo "The greatest number is: " . $num3 . "\n";

    }

} else {

    if ($num2 >= $num3) {

        echo "The greatest number is: " . $num2 . "\n";

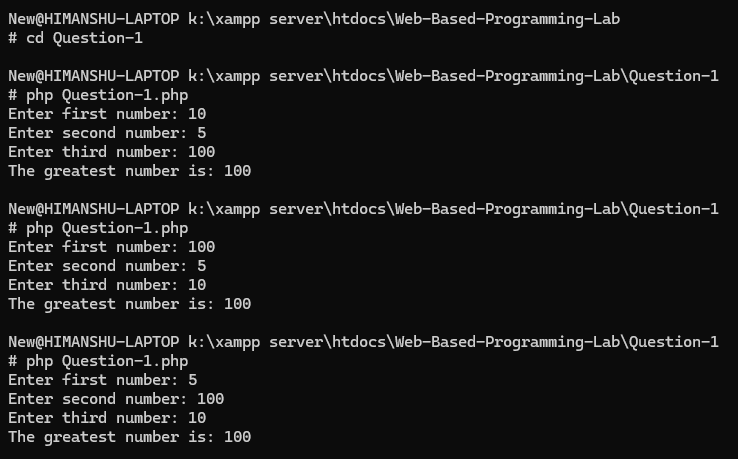
    } else {

        echo "The greatest number is: " . $num3 . "\n";

    }

}

?>



2. WAP to show the usage of switch-case and for/while/do while loop to create a menu driven program for calculator.

Source Code:

<?php

do {

    echo "Calculator Menu:\n";

    echo "1. Addition\n";

    echo "2. Subtraction\n";

    echo "3. Multiplication\n";

    echo "4. Division\n";

    echo "5. Exit\n";

    $choice = readline("Enter your choice: ");

    // Perform calculation based on user choice

    switch ($choice) {

        case 1:

            $num1 = readline("Enter first number: ");

            $num2 = readline("Enter second number: ");

            echo "Result: " . ($num1 + $num2) . "\n";

            break;

        case 2:

            $num1 = readline("Enter first number: ");

            $num2 = readline("Enter second number: ");

            echo "Result: " . ($num1 - $num2) . "\n";

            break;

        case 3:

            $num1 = readline("Enter first number: ");

            $num2 = readline("Enter second number: ");

            echo "Result: " . ($num1 \* $num2) . "\n";

            break;

        case 4:

            $num1 = readline("Enter first number: ");

            $num2 = readline("Enter second number: ");

            if ($num2 == 0) {

                echo "Error: Division by zero\n";

            } else {

                echo "Result: " . ($num1 / $num2) . "\n";

            }

            break;

        case 5:

            echo "Exiting calculator...\n";

            break;

        default:

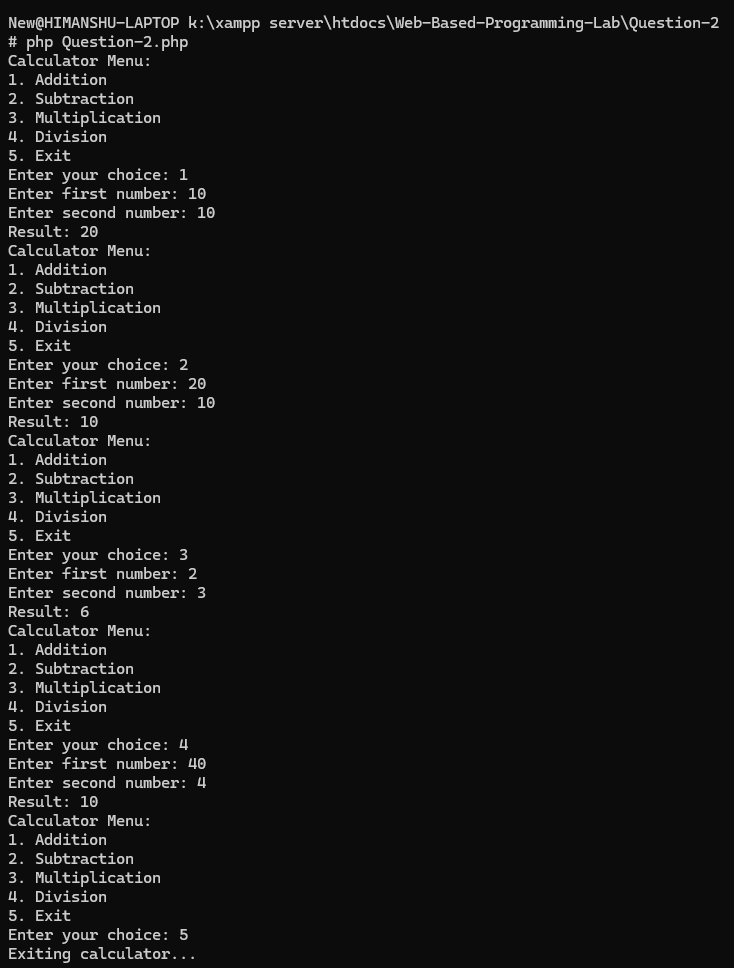
            echo "Invalid choice! Please enter a number between 1 and 5.\n";

            break;

    }

} while ($choice != 5);

?>



3. WAP to implement all the inbuilt array functions like array\_pad(), array\_slice(), array\_splice(), list() functions. (use foreach wherever applicable).

Source Code:

<?php

// Original array

$originalArray = [1, 2, 3, 4, 5];

// array\_pad() - Pad array to the specified length with a value

$paddedArray = array\_pad($originalArray, 7, 0);

echo "Original Array:\n";

foreach($originalArray as $value){

    echo $value . " ";

}

echo "\n";

echo "Array after padding:\n";

foreach ($paddedArray as $value) {

    echo $value . " ";

}

echo "\n\n";

// array\_slice() - Extract a slice of the array

echo "Original Array:\n";

foreach($originalArray as $value){

    echo $value . " ";

}

echo "\n";

$sliceArray = array\_slice($originalArray, 2, 2);

echo "Original Array:\n";

foreach ($sliceArray as $value) {

    echo $value . " ";

}

echo "\n\n";

// array\_splice() - Remove a portion of the array and replace it with something else

$splicedArray = $originalArray;

echo "Original Array:\n";

foreach($originalArray as $value){

    echo $value . " ";

}

echo "\n";

array\_splice($splicedArray, 2, 2, [10, 20]);

echo "Original Array:\n";

foreach ($splicedArray as $value) {

    echo $value . " ";

}

echo "\n\n";

// list() - Assign variables as if they were an array

list($a, $b, $c) = $originalArray;

echo "Original Array:\n";

foreach($originalArray as $value){

    echo $value . " ";

}

echo "\n";

echo "Using list() function:\n";

echo "a = $a, b = $b, c = $c\n";

?>

A screen shot of a computer

Description automatically generated

4. Write a program to perform all four types of sorting.

Source Code:

<?php

// Function to perform bubble sort

function bubbleSort($arr) {

    $n = count($arr);

    for ($i = 0; $i < $n - 1; $i++) {

        for ($j = 0; $j < $n - $i - 1; $j++) {

            if ($arr[$j] > $arr[$j + 1]) {

                $temp = $arr[$j];

                $arr[$j] = $arr[$j + 1];

                $arr[$j + 1] = $temp;

            }

        }

    }

    return $arr;

}

// Function to perform selection sort

function selectionSort($arr) {

    $n = count($arr);

    for ($i = 0; $i < $n - 1; $i++) {

        $minIndex = $i;

        for ($j = $i + 1; $j < $n; $j++) {

            if ($arr[$j] < $arr[$minIndex]) {

                $minIndex = $j;

            }

        }

        $temp = $arr[$i];

        $arr[$i] = $arr[$minIndex];

        $arr[$minIndex] = $temp;

    }

    return $arr;

}

// Function to perform insertion sort

function insertionSort($arr) {

    $n = count($arr);

    for ($i = 1; $i < $n; $i++) {

        $key = $arr[$i];

        $j = $i - 1;

        while ($j >= 0 && $arr[$j] > $key) {

            $arr[$j + 1] = $arr[$j];

            $j = $j - 1;

        }

        $arr[$j + 1] = $key;

    }

    return $arr;

}

// Function to perform quick sort

function quickSort($arr) {

    if (count($arr) <= 1) {

        return $arr;

    }

    $pivot = $arr[0];

    $left = $right = array();

    for ($i = 1; $i < count($arr); $i++) {

        if ($arr[$i] < $pivot) {

            $left[] = $arr[$i];

        } else {

            $right[] = $arr[$i];

        }

    }

    return array\_merge(quickSort($left), array($pivot), quickSort($right));

}

// Sample array to sort

$arr = array(64, 34, 25, 12, 22, 11, 90);

// Bubble Sort

echo "\nBubble Sort:\n";

$bubble\_sort = bubbleSort($arr);

foreach($bubble\_sort as $key => $value) {

    echo "Key: {$key} => Value: {$value}\n";

}

// Selection Sort

echo "\nSelection Sort:\n";

$selection\_sort = selectionSort($arr);

foreach($selection\_sort as $key => $value) {

    echo "Key: {$key} => Value: {$value}\n";

}

// Insertion Sort

echo "\nInsertion Sort:\n";

$insertion\_sort = insertionSort($arr);

foreach($insertion\_sort as $key => $value) {

    echo "Key: {$key} => Value: {$value}\n";

}

// Quick Sort

echo "\nQuick Sort:\n";

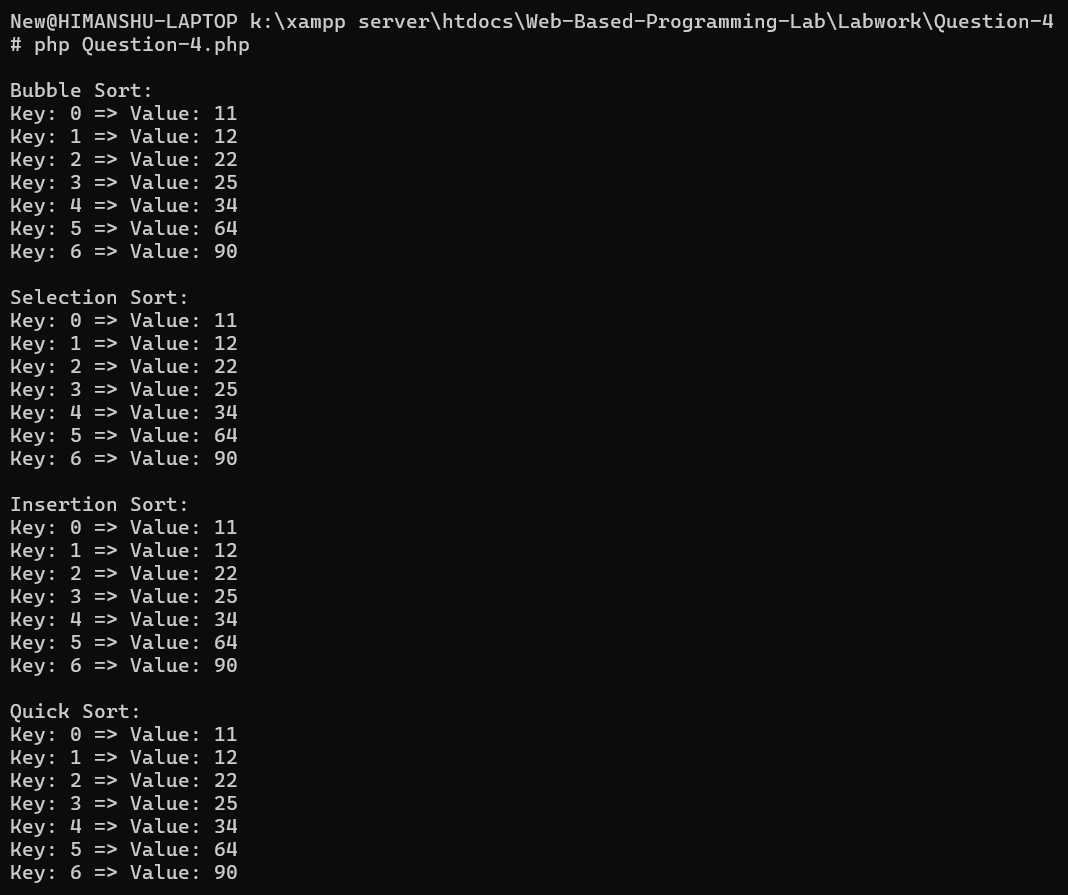
$quick\_sort = quickSort($arr);

foreach($quick\_sort as $key => $value) {

    echo "Key: {$key} => Value: {$value}\n";

}

?>



5. WAP using PHP to implement regular expressions including modifiers, operators and metacharacters.

Source Code:

<?php

// Sample string to search

$string1 = "One 2 three 4 Five, 6";

// Regular expression pattern to match words starting with "fox"

$pattern1 = "/\b[0-9a-zA-Z]+\b/i";

// Performing the regular expression match using preg\_match\_all()

if (preg\_match\_all($pattern1, $string1, $matches)) {

    echo "\nMatches found (Regular expressions - modifiers):\n";

    foreach ($matches[0] as $match) {

        echo $match . "\n";

    }

} else {

    echo "No matches found.\n";

}

//  Sample string to search

$string2 = "

line 1 starts $ here +

line 2 starts here \*

            ";

$pattern2 = "/^line\s.\*+$/m";

// Performing the regular expression match using preg\_match\_all()

if (preg\_match\_all($pattern2, $string2, $matches)) {

    echo "\nMatches found (Regular expressions - operators):\n";

    foreach ($matches[0] as $match){

        echo $match . "\n";

    }

}

else{

    echo "No matches found.\n";

}

// Sample string to search

$string3 = "Helllooo, 123 world, 7890";

$pattern3 = "/\w+,? ?[0-9]\*[0-9]?[0-9]/";

// Performing the regular expression match using preg\_match\_all()

if (preg\_match\_all($pattern3, $string3, $matches)){

    echo "\nMatches found (regular expressions - metacharacters):\n";

    foreach ($matches[0] as $match){

        echo $match . "\n";

    }

}

else{

    echo "No matches found.\n";

}

?>

A computer screen shot of a black screen

Description automatically generated

6. Design a login form and validate using PHP programming. (Use of regular expression, Super global variables such as $\_GET(), $\_POST(), $\_REQUEST() and appropriate functions) .

Login\_form.html:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Login Form</title>

</head>

<body>

    <h2>Login</h2>

    <form action="login.php" method="POST">

        <label for="username">Username:</label><br>

        <input type="text" id="username" name="username" required><br>

        <label for="password">Password:</label><br>

        <input type="password" id="password" name="password" required><br>

        <input type="submit" value="Login">

    </form>

</body>

</html>

login.php

<?php

// Check if the form is submitted

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    // Retrieve username and password from the form

    $username = $\_POST['username'];

    $password = $\_POST['password'];

    // Validate username (pattern: [a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,})

    if (preg\_match("/^[a-z0-9.\_%+-]+@[a-z0-9.-]+\.[a-z]{2,}$/", $username)) {

        // Validate password (min 8 characters, at least one of A-Z, a-z, and 0-9, no spaces)

        if (preg\_match("/^(?=.\*[a-z])(?=.\*[A-Z])(?=.\*\d)[a-zA-Z\d]{8,}$/", $password)) {

            $valid\_username = 'admin@gmail.com';

            $valid\_password = 'Password1234';

            if ($username === $valid\_username && $password === $valid\_password) {

                $user = $username;

                preg\_match("/\b(admin)\b/",$user,$matches);

                $display\_name = $matches[0];

                echo "Welcome " . $display\_name . "!";

                exit;

            } else {

                // Invalid username or password

                echo "Invalid username or password!";

            }

        } else {

            // Invalid password

            echo "Please enter a valid password (min 8 characters, at least one of A-Z, a-z, and 0-9, no spaces)!";

        }

    } else {

        // Invalid username

        echo "Please enter a valid email address!";

    }

} else {

    // If the form is not submitted, redirect back to the login form

    header("Location: login\_form.html");

    exit;

}

?>

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

7. Design a login form and validate using PHP programming (use of PHP Filters).

login\_form.html

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Login Form</title>

</head>

<body>

    <h2>Login</h2>

    <form action="login\_validate.php" method="POST">

        <label for="username">Username:</label><br>

        <input type="text" id="username" name="username" required><br>

        <label for="password">Password:</label><br>

        <input type="password" id="password" name="password" required><br>

        <input type="submit" value="Login">

    </form>

</body>

</html>

login\_validate.php

<?php

// Check if the form is submitted

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

    $username = $\_POST['username'];

    $password = $\_POST['password'];

    // Validate username and password

    if (!empty($username) && !empty($password)) {

        if (filter\_var($username, FILTER\_VALIDATE\_EMAIL)){

        $valid\_username = 'admin@gmail.com';

        $valid\_password = 'password';

        if ($username === $valid\_username && $password === $valid\_password) {

            // Redirect to dashboard or any other page on successful login

            echo "Welcome ".$username."!";

            exit;

        }

else {

            // Invalid username or password

            echo "Invalid username or password!";

        }

        }

    } else {

        // Username or password field is empty

        echo "Please enter both username and password!";

    }

} else {

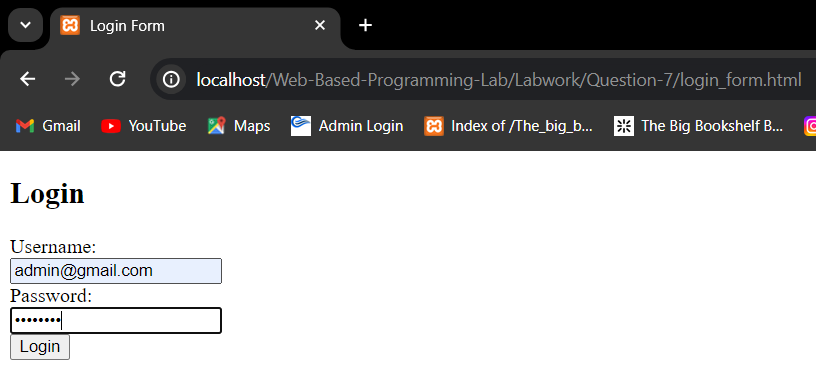
    // If the form is not submitted, redirect back to the login form

    header("Location: login\_form.html");

    exit;

}

?>



A screenshot of a computer

Description automatically generated