SCRIPTING LANGUAGE

LAB EXERCISES

EX.NO:01

IMPLEMENTING STRING AND ARRAY FUNCTIONS IN PHP

AIM:

To write a PHP program to demonstrate five string functions and five array functions.

ALGORITHM:

1.Start the program.

2.Initialize a string variable, perform string operations: length, uppercase, lowercase, reverse, split.

3.Display string results.

4.Initialize an array with numeric values.

5.Perform array operations: count, current element, last element, reverse, sort.

6.Save the file in C:\xamp\htdocs\demo folder for XAMP server, display array results.

7.End the program.

PROGRAM:

```
<?php
// String Functions
$s = "Welcome to PHP";
echo "The given string is: " . $s . "<br>";
// 1. Get the length of the string
$length = strlen($s);
echo "Length of the string: " . $length . "<br>";
// 2. Convert to uppercase
$upper = strtoupper($s);
echo "Uppercase string: " . $upper . "<br>";
// 3. Convert to lowercase
$lower = strtolower($s);
echo "Lowercase string: " . $lower . "<br>";
// 4. Reverse the string
$reversed = strrev($s);
echo "Reversed string: " . $reversed . "<br>";
// 5. Split the string into an array of substrings (length 3)
```

LAB EXERCISES

```
$arr = str_split($s, 3);
echo "Array after splitting:<br>";
print_r($arr);
echo "<br><br>";
// Array Functions
$arr = array(33, 11, 44, 16, 25, 89, 92);
echo "The elements in the array are:<br>";
print_r($arr);
echo "<br>";
// 1. Get the number of elements in the array
$count = count($arr);
echo "Count of elements in the array: " . $count . "<br>";
// 2. Get the current element in the array
$cur = current($arr);
echo "Current element: " . $cur . "<br>";
// 3. Get the last element of the array
$last = end($arr);
echo "Last element: " . $last . "<br>";
// 4. Reverse the array
$reverse = array_reverse($arr);
echo "Reversed array:<br>";
print_r($reverse);
echo "<br>";
// 5. Sort the array in ascending order
sort($arr);
echo "Sorted array:<br>";
print_r($arr);
?>
```

RESULT:

Thus, the above PHP code is executed and the output is obtained.

SCRIPTING LANGUAGE

LAB EXERCISES

EX.NO:02

COLLECTING DATA FROM HTML AND PROCESSING IN PHP

AIM:

To design the HTML form to collect student bio-data and SSLC marks, process the collected data using PHP, and calculate total and average marks.

ALGORITHM:

1.Start the program.

2.Create an HTML form to collect student name, age, and marks for 5 subjects.

3.Submit the form data to a PHP script using the POST method.

4.In PHP, collect the student name and age.

5.Collect the marks into an array, calculate total and average.

6.Display the student name, age, total marks, and average marks.

7.End the program.

PROGRAM:

1) HTML form to collect student bio-data and SSLC marks:

HTML Form – EX-NO-02SCRIPTING.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Student Biodata Form</title>
</head>
<body>
<h2>Student Biodata and SSLC Marks</h2>
<form action="EX-NO-02SCRIPTING.php" method="POST">
<label for="name">Name:</label>
<input type="text" id="name" name="name" required><br><br>
<label for="age">Age:</label>
<input type="number" id="age" name="age" required min="15"><br><br>
```

LAB EXERCISES

```
<h3>SSLC Marks</h3>
<label for="s1">Subject 1:</label>
<input type="number" id="s1" name="marks[]" required min="0" max="100"><br><br>
<label for="s2">Subject 2:</label>
<input type="number" id="s2" name="marks[]" required min="0" max="100"><br><br>
<label for="s3">Subject 3:</label>
<input type="number" id="s3" name="marks[]" required min="0" max="100"><br><br>
<label for="s4">Subject 4:</label>
<input type="number" id="s4" name="marks[]" required min="0" max="100"><br><br>
<label for="s5">Subject 5:</label>
<input type="number" id="s5" name="marks[]" required min="0" max="100"><br><br>
<input type="submit" value="Submit">
</form>
</body>
</html>
```

2) Processing collected data using PHP:

PHP Script – EX-NO-02SCRIPTING.php

```
<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
   // Collect biodata
   $name = htmlspecialchars($_POST['name']);
   $age = (int)$_POST['age'];
   // Collect marks and calculate total and average
   $marks = $_POST['marks'];
   $total = array_sum($marks);
   $average = $total / count($marks);
   // Display results
   echo "<h2>Student Biodata</h2>";
   echo "Name: $name<br>";
   echo "Age: $age<br>";
   echo "Total Marks: $total<br>";
   echo "Average Marks: " . number_format($average, 2) . "<br>";
```

```
} else {
   echo "Invalid request method.";
}
?>
```

RESULT:

Thus the above HTML and PHP code are executed and the output is obtained.

EX.NO:03

USING PHP AND MYSQL DATABASE

AIM:

To develop a simple PHP application which displays the result of the student by entering their register number as user input(assuming student marks are already available in the MYSQL database).

ALGORITHM:

1.Start the program.

2.Create a MySQL database college and a table student with columns: Reg_No, Name, M1 to M5.

3.Insert student data into the student table.

4.Create an HTML form to take register number as input.

5.In PHP, connect to the MySQL database.

6.Retrieve student details based on the entered register number, calculate total and average marks.

7.Display the student name, register number, marks, total, and average. End the program.

PROGRAM:

1) Database and Table creation in MYSQLPHP:

- Creating database 'College':
  - mysql> CREATE DATABASE College;
  - Query OK, 1 row affected (0.02 sec)
- Showing databases:
  - mysql> show databases;
  - Query Ok, 6 rows in set (0.01 sec)
- Using 'College' database:
  - mysql> use College;
  - Database changed
- Creating 'Student' table

LAB EXERCISES

- mysql> CREATE TABLE Student (ID INT AUTO_INCREMENT PRIMARY KEY, Reg_No VARCHAR(50), Name VARCHAR(100), M1 INT, M2 INT, M3 INT, M4 INT, M5 INT);
- Query OK, 0 rows affected (0.03 sec)

- Inserting rows into table 'Student':
  - mysql> INSERT INTO Student (Reg_No,Name,M1,M2,M3,M4, M5) VALUES
  - ('COMP1111', 'Lanish', 78, 89, 90, 98, 99),
  - ('COMP1112', 'Jerin', 88,99,90,67,76),
  - ('COMP1113', 'Merin', 97,98,96,95,90),
  - ('COMP1114', 'Janan', 78,89,83,99,93),
  - ('COMP1115', 'Subbu', 67,78,89,90,94);
  - Query OK, 5 rows affected (0.00 sec)
  - Records: 5 Duplicates: 0 Warnings: 0
- Showing Student table contents:
  - mysql> SELECT * FROM Student;
  - 5 rows in set (0.00 sec)

2) HTML Form for user input:

HTML Form – EX-NO-03SCRIPTING.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Student Result Display</title>
</head>
<body>
<h2>Student Result</h2>
<form action="EX-NO-03SCRIPTING.php" method="POST">
<label for="Reg_No">Enter Register Number:</label>
<input type="text" id="Reg_no" Name="Reg_No" required>
<input type="submit" value="Get Result">
</form>
</body>
</html>
```

3) PHP Code to Retrive and Display results:

PHP Script – EX-NO-03SCRIPTING.php

```
<?php
// Database connection parameters
$host = 'localhost';
$db = 'college';
$user = 'root';
$pass = '';
// Create connection
$conn = new mysqli($host, $user, $pass, $db);
// Check connection
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
```

LAB EXERCISES

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {
   $Reg_No = $_POST["Reg_No"];
   // Query to get student data
   $sql = "SELECT * FROM student WHERE Reg_No = '$ Reg_No ";
   $result = $conn->query($sql);
   if ($result->num_rows > 0) {
      $row = $result->fetch_assoc();
      $total = $row['M1'] + $row['M2'] + $row['M3'] + $row['M4'] + $row['M5'];
      $average = $total / 5;
      // Display student result
      echo "<h2>Student Result</h2>";
      echo "Name: " . $row['name'] . "<br>";
      echo "Register Number: " . $row['reg_no'] . "<br>";
      echo "Marks:<br>";
      echo "Subject 1: " . $row['M1'] . "<br>";
      echo "Subject 2: " . $row['M2'] . "<br>";
      echo "Subject 3: " . $row['M3'] . "<br>";
      echo "Subject 4: " . $row['M4'] . "<br>";
      echo "Subject 5: " . $row['M5'] . "<br>";
      echo "Total Marks: " . $total . "<br>";
      echo "Average Marks: " . number_format($average, 2) . "<br>";
   } else {
      echo "No student found with the given registration number.";
   }
}
// Close the connection
$conn->close();
?>
```

RESULT:

Thus the above HTML and PHP code are executed and the output is obtained.

EX.NO:04

VALIDATING USERNAME AND PASSWORD USING PHP

AIM:

To develop a simple login page that validates the username and password from the database. If valid, the user is redirected to welcome.php displaying their name; otherwise, the login page remains.

ALGORITHM:

1.Start the program.
2.Create a MySQL table users in the college database with columns: username, password, sname.
3.Insert sample users into the table.
4.Create an HTML login form to collect username and password.
5.In PHP, connect to the database and validate the entered credentials.

LAB EXERCISES

6.If valid, start a session, store the student name, and redirect to welcome.php.
7.If invalid, show an error message on the login page, End the program.

PROGRAM:

1) Table creation in MYSQLPHP:

- **Showing databases:**
  - mysql> show databases;
  - Query Ok, 6 rows in set (0.01 sec)
- **Using 'College' database:**
  - mysql> USE College;
  - Database changed
- **Creating 'Users' table:**
  - CREATE TABLE Users(ID INT AUTO_INCREMENT PRIMARY KEY, UserName VARCHAR(50) UNIQUE, Password VARCHAR(20), Name VARCHAR(50));
  - Query OK, 0 rows affected (0.05 sec)
- **Inserting rows into table 'Users':**
  - mysql> INSERT INTO Users(UserName, Password, Name) VALUES
  - ('KPTC1','comp2003@','Malar Kannan'),
  - ('KPTC2','comp2004@','Kumar'),
  - ('KPTC3', 'comp2005@', 'Prithiv');
  - Query OK, 3 rows affected (0.00 sec)
  - Records: 3 Duplicates: 0 Warnings: 0
- **Showing 'Users' table contents:**
  - mysql> select from users;
  - 3 rows in set(0.00 sec)

2) HTML code for Login page:

HTML Login Page – EX-NO-04SCRIPTING.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Login Page</title>
</head>
<body>
<h2>Login</h2>
```

LAB EXERCISES

```
<form action="EX-NO-04SCRIPTING.php" method="POST">
  <label for="UserName">UserName:</label>
  <input type="text" id="username" name="username" required><br><br>
  <label for="Password">Password:</label>
  <input type="Password" id="Password" name="Password" required><br><br>
  <input type="submit" value="Login">
</form>
</body>
</html>
```

3) PHP code for getting user data and redirecting to welcome page:

PHP Script to Validate Login – EX-NO-04SCRIPTING.php

```
<?php
session_start();
// Database connection
$host = 'localhost';
$db = 'college';
$user = 'root';
$pass = '';
$conn = new mysqli($host, $user, $pass, $db);
if ($conn->connect_error) {
   die("Connection failed: " . $conn->connect_error);
}
if ($_SERVER["REQUEST_METHOD"] == "POST") {
   $username = $_POST['username'];
   $password = $_POST['password'];
   // Query to check user
   $sql = "SELECT * FROM users WHERE username='$username' AND password='$password'";
   $result = $conn->query($sql);
   if ($result->num_rows > 0) {
      $row = $result->fetch_assoc();
      $_SESSION['sname'] = $row['sname'];
      header("Location: welcome.php");
      exit();
   } else {
      echo "Invalid username or password. Please try again.";
   }
}
$conn->close();
?>
```

4) PHP Code for Welcome Page – welcome.php:

```
<?php
session_start();
$sname = $_SESSION['sname'];
?>
<!DOCTYPE html>
```

LAB EXERCISES

```
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Welcome</title>
</head>
<body>
<h2>Welcome, <?php echo $sname; ?>!</h2>
<p>You have successfully logged in.</p>
</body>
</html>
```

RESULT:

Thus the above HTML and PHP code are executed and the output is obtained.

EX.NO:05

USING JQUERY

AIM:

To write a code using jQuery to disable the right-click (context menu) option on a webpage.

ALGORITHM:

1.Start the program.
2.Create an HTML document structure with <html>, <head>, and <body> tags.
3.Include the <head> section in the HTML file.
4.Add meta tags for character set (UTF-8) and viewport for responsive design.
5.Include the jQuery library using a CDN link inside the <head> section.
6.Add a <script> tag to write the jQuery code.
7.Use $(document).ready() to ensure the DOM is fully loaded before running the script.
8.Detect the right-click event using the contextmenu event in jQuery.
9.Prevent the default right-click menu using event.preventDefault().
10.Display an alert message when the user attempts to right-click.
11.End the program.

PROGRAM:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Disable Right Click</title>

  <!-- jQuery Library -->
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
```

LAB EXERCISES

```
  <script>
    $(document).ready(function () {
      $(document).on("contextmenu", function (e) {
        e.preventDefault();
        alert("Right-click is disabled on this webpage.");
      });
    });
  </script>
</head>
<body>
  <h1>Right Click Disabled Example</h1>
  <p>Try right-clicking anywhere on this page!</p>
</body>
</html>
```

RESULT:

Thus, the above program using jQuery is executed successfully, and the right-click option is disabled on the webpage.

EX.NO:06

USING AJAX

AIM:

To develop a simple application which displays details of the college by getting college code as input using AJAX without reloading the page (assuming college details like code, name, courses_offered, address, hostel facility, etc., are already available in the database).

ALGORITHM:

1.Start the program.
2.Create a MySQL database named college.
3.Create a table college with the required fields.
4.Insert sample college records into the table.
5.Create an HTML page to accept the college code from the user.
6.Design the input form and result display area using HTML and CSS.
7.Use JavaScript and AJAX to send the entered college code to the server.
8.Create a PHP file to receive the AJAX request.
9.Retrieve college details from the database using the given college code.
10.Return the fetched data in JSON format.
11.Display the result on the web page without reloading it.
12.End the program.

SCRIPTING LANGUAGE

LAB EXERCISES

PROGRAM:

1) Database and Table Creation (MySQL):

- **Using 'College' database:**
  - mysql> USE College;
  - Database changed

- **Creating 'Colleges' table:**

  - CREATE TABLE Colleges (Code INT PRIMARY KEY, Name VARCHAR(50), Courses_Offered TEXT, Address VARCHAR(50), Hostel_Facility BOOLEAN);
  - Query OK, 0 rows affected (0.05 sec)

- **Inserting rows into table 'Colleges':**

  - mysql> INSERT INTO colleges VALUES
  - (1001, 'ABC Polytechnic College', 'Civil, Mech, EEE, ECE, Computer', 'Trichy', TRUE),
  - (1002, 'XYZ Polytechnic College', 'Civil, Mech, EEE, Computer, Auto', 'Madurai', TRUE),
  - (1003, 'Vetti Polytechnic College', 'Mech, Biomedical, EEE, Computer', 'Nagercoil', FALSE);

- **Showing 'Colleges' table contents:**

  - mysql> SELECT * FROM Colleges;
  - rows in set (0.00 sec)

2) HTML + AJAX Code:

File Name: EX-NO-06SCRIPTING.html

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>College Details</title>
<style>
body {
font-family: Arial;
margin: 20px;
}
.result {
margin-top: 20px;
}
.error {
color: red;
}
</style>
</head>
<body>
```

LAB EXERCISES

```
<h1>Get College Details</h1>
<form id="cform">
<label>Enter College Code:</label>
<input type="text" id="ccode" required>
<button type="submit">Get Details</button>
</form>
<div id="result" class="result"></div>
<script>
document.getElementById("cform").addEventListener("submit", function(e) {
e.preventDefault();
let ccode = document.getElementById("ccode").value;
let resultDiv = document.getElementById("result");
resultDiv.innerHTML = "";
let xhr = new XMLHttpRequest();
xhr.open("GET", "EX-NO-06SCRIPTING.php?ccode=" + ccode, true);
xhr.onreadystatechange = function () {
if (xhr.readyState === 4 && xhr.status === 200) {
let response = JSON.parse(xhr.responseText);
if (response.error) {
resultDiv.innerHTML = "<p class='error'>" + response.error + "</p>";
} else {
resultDiv.innerHTML = `
<h2>${response.name}</h2>
<p><strong>Code:</strong> ${response.code}</p>
<p><strong>Courses Offered:</strong> ${response.courses_offered}</p>
<p><strong>Address:</strong> ${response.address}</p>
<p><strong>Hostel Facility:</strong>
${response.hostel_facility == 1 ? "Available" : "Not Available"}</p>
`;
}
}
};
xhr.send();
});
</script>
</body>
</html>
```

3) PHP Code:

File Name: EX-NO-06SCRIPTING.php

```
<?php
header("Content-Type: application/json");
$host = "localhost";
$user = "root";
$pass = "";
$db = "college";
$conn = new mysqli($host, $user, $pass, $db);
if ($conn->connect_error) {
```

LAB EXERCISES

```
    echo json_encode(["error" => "Database connection failed"]);
    exit;
}
$ccode = $_GET['ccode'];
$sql = "SELECT * FROM colleges WHERE code = '$ccode'";
$result = $conn->query($sql);
if ($result->num_rows > 0) {
    echo json_encode($result->fetch_assoc());
} else {
    echo json_encode(["error" => "No college found with the given code"]);
}
$conn->close();
?>
```

RESULT:

Thus, the above HTML and PHP code AJAX application is executed successfully and the output is obtained.

EX.NO:07

UPLOADING FILE TO SERVER USING NODE.JS

AIM:

To develop a Node.js application to upload a file from the client and store it on the server using the Formidable module.

ALGORITHM:

1.Start the program.
2.Install the required formidable module using NPM.
3.Import required Node.js modules such as http, fs, and formidable.
4.Create an HTTP server using Node.js.
5.Display a file upload form when the server is accessed.
6.Accept the uploaded file from the client.
7.Store the uploaded file temporarily on the server.
8.Move the uploaded file to the required directory.
9.Display a success message after the file is uploaded.
10.Stop the program.

PROCEDURE:

1.Open the Node.js Command Prompt.
2.Install the formidable module using the command:
3.npm install formidable
4.Open Notepad and type the Node.js code.
5.Save the file as EXNO7.js in the Node.js folder (Example: C:\Users\Student).
6.Run the program using the command:
7.node EXNO7.js

LAB EXERCISES

8.Open a web browser and type the URL:
9.http://localhost
10.Choose a file and click Submit.

PROGRAM:

File Name: EX-NO-07.js

```
var http = require('http');
var formidable = require('formidable');
var fs = require('fs');
http.createServer(function (req, res) {
if (req.url == '/fileupload' && req.method.toLowerCase() == 'post') {
var form = new formidable.IncomingForm();
form.parse(req, function (err, fields, files) {
var oldpath = files.filetoupload[0].filepath;
var newpath = 'C:/Users/Suresh/' + files.filetoupload[0].originalFilename;
fs.rename(oldpath, newpath, function (err) {
if (err) throw err;
res.write('File uploaded and moved!');
res.end();
});
});
} else {
res.writeHead(200, { 'Content-Type': 'text/html' });
res.write('<form action="fileupload" method="post" enctype="multipart/form-data">');
res.write('<input type="file" name="filetoupload"><br><br>');
res.write('<input type="submit">');
res.write('</form>');
res.end();
}
}).listen(8080);
console.log("Server running at http://localhost");
```

RESULT:

Thus, the Node.js program to upload a file to the server using the Formidable module is executed successfully, and the uploaded file is stored in the server directory.

EX.NO:08

SENDING EMAIL USING NODE.JS

AIM:

To develop a Node.js program to send an email using the Node mailer module and a Gmail account with App Password authentication.

ALGORITHM:

1)Install Node.js and the Nodemailer module.
2)Open Command Prompt and run the command:

LAB EXERCISES

3)npm install nodemailer
4)Enable 2-Step Verification for your Gmail account (if not already enabled).
5)Generate an App Password from your Google Account settings to be used instead of your regular password.
6)Write the Node.js code to send an email using Nodemailer, with your Gmail address and the generated App Password.
7)Run the Node.js script from the Command Prompt using:
8)node EX-NO-08.js
9)Check the recipient's email inbox to confirm that the email has been received.
10)End the program.

PROCEDURE:

1)Download and install the Nodemailer module using NPM in the Node.js Command Prompt.
2)C:\Users\Suresh> npm install nodemailer
3)Enter the Node.js code in Notepad and save it as EXNO8.js in your Node.js folder (Example: C:\Users\Student).
4)Generate an App Password (for Gmail users):

i)Go to your Google Account → Security Settings.
ii)Enable 2-Step Verification if it is not already enabled.
iii)Under the Signing in to Google section, click App Passwords.
iv)Generate a password for your Node.js application.
v)Use this generated password in the auth.pass field in the Node.js code.

PROGRAM:

```
const nodemailer = require('nodemailer');
const transporter = nodemailer.createTransport({
service: 'gmail',
auth: {
user: 'your-email@gmail.com', // replace with your Gmail address
pass: 'your-app-password'     // replace with your generated App Password
}
});
const mailOptions = {
from: 'your-email@gmail.com',
to: 'recipient-email@gmail.com', // replace with recipient email
subject: 'Test Email from Node.js',
text: 'This is a test email sent using Node.js!',
html: '<b>This is a test email sent using Node.js!</b>'
};
transporter.sendMail(mailOptions, function(error, info){
if(error){
console.log('Error occurred:', error.message);
} else {
console.log('Email sent successfully:', info.response);
}
});
```

RESULT:

Thus, the above Node.js program is executed and the output is obtained successfully.