Module 1: JavaScript and PHP Tasks

1. JavaScript Code

a. Demonstrating Different JavaScript Objects (String, RegExp, Math, Date)

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Practical 1a</title>
<script>
    var str = "JavaScript is fun. JavaScript is powerful.";
    var regex = /JavaScript/g;
    var num = 25.89;
    var currentDate = new Date();
    document.write("<h3>String Object Example:</h3>");
    document.write("<br>Length of String: " + str.length);
    document.write("<br>Substring (from index 19): " + str.substring(19));
    document.write("<h3>RegExp Object Example:</h3>");
    document.write("<br>Does the text contain 'JavaScript'? " + regex.test(str));
    var replacedText = str.replace(regex, "JS");
    document.write("<br>Text after replacement: " + replacedText);
    document.write("<h3>Math Object Example:</h3>");
    document.write("<br>Floor (round down): " + Math.floor(num));
    document.write("<br>Ceil (round up): " + Math.ceil(num));
    document.write("<br>Round (nearest integer): " + Math.round(num));
    document.write("<br>Square root: " + Math.sqrt(num));
    document.write("<br>Power (2^3): " + Math.pow(2, 3));
    document.write("<h3>Date Object Example:</h3>");
    document.write("<br>Current Date and Time: " + currentDate);
    document.write("<br>Year: " + currentDate.getFullYear());
</script>
</head>
<body>
</body>
</html>
```

b. Demonstrating Different JavaScript Objects (Window, Navigator, History, Location, Document)

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>JavaScript Objects Example</title>
</head>
<body>
<h3>JavaScript Objects Example</h3>
<script>
        // Alert the user with a welcome message
        alert("Welcome to the JavaScript Objects Example!");

        // Collecting information from different JavaScript objects
        document.write("<strong>Window Object:</strong><br>");
        document.write("Inner Width: " + window.innerWidth + "px<br>");
        document.write("Inner Height: " + window.innerHeight + "px<br>");
        document.write("Outer Width: " + window.outerWidth + "px<br>");
        document.write("Outer Height: " + window.outerHeight + "px<br><br>");
        document.write("<strong>Navigator Object:</strong><br>");
        document.write("User  Agent: " + navigator.userAgent + "<br>");
        document.write("Platform: " + navigator.platform + "<br><br>");
        document.write("<strong>History Object:</strong><br>");
        document.write("Number of entries in history: " + history.length + "<br>");
        document.write("<button onclick='history.back()'>Go Back</button> ");
        document.write("<button onclick='history.forward()'>Go Forward</button><br><br>");
        document.write("<strong>Location Object:</strong><br>");
        document.write("Current URL: " + location.href + "<br>");
        document.write("Protocol: " + location.protocol + "<br>");
        document.write("Host: " + location.host + "<br>");
        document.write("<button onclick=\"location.href='https://www.amazon.in'\">Go to
    Example.com</button><br><br>");
        document.write("<strong>Document Object:</strong><br>");
        document.write("Document Title: " + document.title + "<br>");
        document.write("Document URL: " + document.URL + "<br>"); // Example of Document Object


        // Change background color
        document.body.style.backgroundColor = 'lightblue';
</script>
</body>
</html>
```

c. Storing and Retrieving Cookies

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Cookie Example</title>
</head>
<body>
<h1>Cookie Example</h1>
<input type="text" id="cookieName" placeholder="Cookie Name">
<input type="text" id="cookieValue" placeholder="Cookie Value">
<button onclick="setCookie()">Set Cookie</button>
<button onclick="getCookieValue()">Get Cookie</button>
<p id="output"></p>

<script>
        // Function to set a cookie
        function setCookie() {
            const name = encodeURIComponent(document.getElementById('cookieName').value);
            const value = encodeURIComponent(document.getElementById('cookieValue').value);
            if (!name || !value) {
                alert('Please provide both a cookie name and value.');
                return;
            }
            document.cookie = `${name}=${value}; path=/`;
            alert('Cookie has been set!');
        }
        function getCookieValue() {
            const name = encodeURIComponent(document.getElementById('cookieName').value) ;
            document.getElementById('output').textContent = name
                ? `Cookie Value: ${name}`
                : 'Cookie not found!';
        }
</script>
</body>
</html>
```

2. **XML File with Internal/External DTD and Displaying Using CSS/XSL**

a. **XML File with Internal DTDusing CSS**

catalog.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="catalog.css"?>
<!DOCTYPE catalog [
<!ELEMENT catalog (book+)>
<!ELEMENT book (title, author, year, price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT price (#PCDATA)>
]>
<catalog>
<book>
<title>Harry Potter and the Sorcerer's Stone</title>
<author>J.K. Rowling</author>
<year>1997</year>
<price>19.99</price>
</book>
<book>
<title>The Hobbit</title>
<author>J.R.R. Tolkien</author>
<year>1937</year>
<price>15.99</price>
</book>
</catalog>
```

catalog.css

```
catalog {
   font-family: Arial, sans-serif;
   margin: 20px;
}
book {
   margin: 15px 0;
   padding: 10px;
   border-bottom: 1px solid lightgray;
}
title {
   font-weight: bold;
   color: teal;
}
author {
   font-style: italic;
   color: coral;
}
year {
    font-size: 0.9em;
    color: charcoal;
}
price {
    color: charcoal;
    font-weight: bold;
}
```

b. **XML File with External DTDusing CSS**

catalog.dtd

```
<!ELEMENT catalog (book+)>
<!ELEMENT book (title, author, year, price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT price (#PCDATA)>
```

catalog.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/css" href="catalog.css"?>
<!DOCTYPE catalog SYSTEM "catalog.dtd">
<catalog>
<book>
<title>Harry Potter and the Sorcerer's Stone</title>
<author>J.K. Rowling</author>
<year>1997</year>
<price>19.99</price>
</book>
<book>
<title>The Hobbit</title>
<author>J.R.R. Tolkien</author>
<year>1937</year>
<price>15.99</price>
</book>
</catalog>
```

catalog.css

```
catalog {
    font-family: Arial, sans-serif;
    margin: 20px;
}
book {
    margin: 15px 0;
    padding: 10px;
    border-bottom: 1px solid lightgray;
}
title {
    font-weight: bold;
    color: teal;
}
author {
    font-style: italic;
    color: coral;
}
year {
    font-size: 0.9em;
    color: charcoal;
}
price {
    color: charcoal;
    font-weight: bold;
}
```

c. **XML File with Internal DTDusing XSL**

catalog.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="catalog.xsl"?>
<!DOCTYPE catalog [
<!ELEMENT catalog (book+)>
<!ELEMENT book (title, author, year, price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT price (#PCDATA)>
]>
<catalog>
<book>
<title>Harry Potter and the Sorcerer's Stone</title>
<author>J.K. Rowling</author>
<year>1997</year>
<price>19.99</price>
</book>
<book>
<title>The Hobbit</title>
<author>J.R.R. Tolkien</author>
<year>1937</year>
<price>15.99</price>
</book>
</catalog>
```

catalog.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
```

```
<!-- Match the root element (catalog) -->
<xsl:template match="/catalog">
<html>
<head>
<title>Book Catalog</title>
<style>
        body {
          font-family: Arial, sans-serif;
          margin: 20px;
          padding: 20px;
          background-color: ghostwhite;
        }
        h2 {
          color: #2c3e50;
        }
        .book {
          margin: 10px 0;
          padding: 10px;
          background-color: white;
          border: 1px solid lightgray;
          border-radius: 5px;
        }
        .book h3 {
          color: brightblue;
        }
</style>
</head>
<body>
<h2>Catalog of Books</h2>
<xsl:for-each select="book">
<div class="book">
<h3><xsl:value-of select="title"/></h3>
<p><strong>Author:</strong><xsl:value-of select="author"/></p>
<p><strong>Year:</strong><xsl:value-of select="year"/></p>
<p><strong>Price:</strong> $<xsl:value-of select="price"/></p>
</div>
</xsl:for-each>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

d. **XML File with External DTDusing XSL**

catalog.dtd

```
<!ELEMENT catalog (book+)>
<!ELEMENT book (title, author, year, price)>
```

```
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT year (#PCDATA)>
<!ELEMENT price (#PCDATA)>
```

catalog.xml

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="catalog.xsl"?>
<!DOCTYPE catalog SYSTEM "catalog.dtd">
<catalog>
<book>
<title>Harry Potter and the Sorcerer's Stone</title>
<author>J.K. Rowling</author>
<year>1997</year>
<price>19.99</price>
</book>
<book>
<title>The Hobbit</title>
<author>J.R.R. Tolkien</author>
<year>1937</year>
<price>15.99</price>
</book>
</catalog>
```

catalog.xsl

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

<!-- Match the root element (catalog) -->
<xsl:template match="/catalog">
<html>
<head>
<title>Book Catalog</title>
<style>
        body {
          font-family: Arial, sans-serif;
          margin: 20px;
          padding: 20px;
          background-color: ghostwhite;
        }
        h2 {
          color: #2c3e50;
        }
        .book {
          margin: 10px 0;
          padding: 10px;
          background-color: white;
          border: 1px solid lightgray;
          border-radius: 5px;
        }
        .book h3 {
          color: brightblue;
        }
</style>
</head>
<body>
<h2>Catalog of Books</h2>
<xsl:for-each select="book">
<div class="book">
<h3><xsl:value-of select="title"/></h3>
<p><strong>Author:</strong><xsl:value-of select="author"/></p>
<p><strong>Year:</strong><xsl:value-of select="year"/></p>
<p><strong>Price:</strong> $<xsl:value-of select="price"/></p>
</div>
</xsl:for-each>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

3. PHP Scripts for Mathematical Operations

a. Calculating Factorial

```
<?php
function factorial($n) {
   if ($n <= 1) return 1;
   return $n * factorial($n - 1);
}
$number = 5;
echo "Factorial of $number is: " . factorial($number)     // Output: 120
?>
```

b. **Fibonacci Series**

```
<?php
function fibonacci($n) {
   if ($n <= 1) return $n;
   return fibonacci($n - 1) + fibonacci($n - 2);
}

for ($i = 0; $i < 10; $i++) {
   echo fibonacci($i) . " ";
}
?>
```

c. **Displaying Prime Numbers in a Given Range**

```
<?php
function is_prime($n) {
    if ($n < 2) return false;
    for ($i = 2; $i <= sqrt($n); $i++) {
        if ($n % $i == 0) return false;
    }
    return true;
}

for ($i = 1; $i <= 20; $i++) {
    if (is_prime($i)) {
        echo $i . " ";
    }
}
?>
```

d. **Evaluating Expressions**

```
<?php
$expression = "3 + 4 * 2";
echo eval("return $expression;");  // Output: 11
?>
```

4. PHP Scripts for Working with Forms, Arrays, and Files

a. Retrieving Data from HTML Forms

```
<form method="post" action="process.php">
    Name: <input type="text" name="name">
<input type="submit">
</form>

<?php
// process.php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST['name'];
    echo "Hello, " . $name;
}
?>
```

b. **Working with Arrays**

```
<?php
$fruits = array("Apple", "Banana", "Orange");
echo $fruits[1];  // Output: Banana
?>
```

c. **Working with Files (Reading/Writing)**

```
<?php
// Writing to a file
```

```
$file = fopen("example.txt", "w");
fwrite($file, "Hello, this is a test.");
fclose($file);

// Reading from a file
$file = fopen("example.txt", "r");
echo fread($file, filesize("example.txt"));
fclose($file);
?>
```

5. Advanced PHP

a. Demonstrating Use of Sessions and Cookies

```
<?php
// Starting a session
session_start();
$_SESSION["user"] = "Gufran";

// Setting a cookie
setcookie("user", "Gufran", time() + 3600);  // 1 hour expiry
echo "Logged in as " . $_SESSION["user"];
session_destroy();
?>
```

b. **Demonstrating Use of Filters**

```
<?php
$email = "user@domain.com";
if (filter_var($email, FILTER_VALIDATE_EMAIL)) {
    echo "Valid email.";
} else {
    echo "Invalid email.";
}
?>
```

Module 2

6. PHP and MySQL

a. Write a PHP program to create: Create a database College

```
<?php
$servername = "localhost";
$username = "root";
$password = ""; // Update your password if necessary

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

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

// Create database College
$sql = "CREATE DATABASE College";
if ($conn->query($sql) === TRUE) {
    echo "Database College created successfully.<br>";
} else {
    echo "Error creating database: " . $conn->error;
}

$conn->close();
?>
```

b. Create a table Department (Dname, Dno, Number_of_faculty)

```
<?php
$servername = "localhost";
$username = "root";
$password = ""; // Update your password if necessary

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

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

$conn->select_db("College");
// Create table Department
$sql = "CREATE TABLE Department (
    Dname VARCHAR(50),
    Dno INT,
    Number_of_faculty INT
)";
if ($conn->query($sql) === TRUE) {
    echo "Table Department created successfully.<br>";
} else {
    echo "Error creating table: " . $conn->error;
}

$conn->close();
?>
```

c. Write a PHP program to create a database named “College”. Create a table named “Student” with following fields (sno, sname, percentage). Insert 3 records of your choice. Display the names of the students whose percentage is between 35 to 75 in a tabular format.

```
<?php
$servername = "localhost";
$username = "root";
$password = ""; // Update your password if necessary
$dbname = "College";

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

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

// Create table Student
$sql = "CREATE TABLE Student (
    sno INT AUTO_INCREMENT PRIMARY KEY,
    sname VARCHAR(50),
    percentage FLOAT
)";
if ($conn->query($sql) === TRUE) {
    echo "Table Student created successfully.<br>";
} else {
    echo "Error creating table: " . $conn->error;
}

// Insert 3 records
$sql = "INSERT INTO Student (sname, percentage) VALUES
        ('John', 45.5),
        ('Alice', 67.8),
        ('Mark', 28.4)";
if ($conn->query($sql) === TRUE) {
    echo "Records inserted successfully.<br>";
} else {
    echo "Error inserting records: " . $conn->error;
}

// Display names of students with percentage between 35 and 75
$sql = "SELECT sname FROM Student WHERE percentage BETWEEN 35 AND 75";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    echo "<table border='1'><tr><th>Student Name</th></tr>";
    while ($row = $result->fetch_assoc()) {
        echo "<tr><td>" . $row['sname'] . "</td></tr>";
    }
    echo "</table>";
} else {
    echo "No records found.";
}

$conn->close();
?>
```

7. Write a PHP program

a. Update rows in a table

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

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

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

// Update student percentage
$sql = "UPDATE Student SET percentage = 55.5 WHERE sname = 'Mark'";
if ($conn->query($sql) === TRUE) {
    echo "Record updated successfully.<br>";
} else {
    echo "Error updating record: " . $conn->error;
}

$conn->close();
?>
```

c. Delete rows from a table

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

    // Create connection
```

```
$conn = new mysqli($servername, $username, $password, $dbname);

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

// Delete a student record
$sql = "DELETE FROM Student WHERE sname = 'Alice'";
if ($conn->query($sql) === TRUE) {
   echo "Record deleted successfully.<br>";
} else {
   echo "Error deleting record: " . $conn->error;
}

$conn->close();
?>
```

8. Design a PHP page for authenticating a user

```
<?php
session_start();
// Hardcoded user credentials (for demo purposes)
$users = ['admin' => 'password123', 'user1' => 'pass123'];
// Handle login
if ($_SERVER['REQUEST_METHOD'] === 'POST') {
   if (isset($users[$_POST['username']]) && $users[$_POST['username']] === $_POST['password'])
{
      $_SESSION['username'] = $_POST['username'];
      echo "Welcome, " . htmlspecialchars($_POST['username']) . "!";
      exit;
   } else {
      echo "Invalid username or password.";
   }
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Login</title>
</head>
<body>
   <form method="POST">
      <input type="text" name="username" placeholder="Username" required>
      <input type="password" name="password" placeholder="Password" required>
```

```
        <button type="submit">Login</button>
    </form>
</body>
</html>
```