Module 1: JavaScript and PHP Tasks

1. JavaScript Code

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

```html
<!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)

```html
<!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
    Amazon.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
```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
```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
<?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>
```

```xml
<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
```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
<?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
<?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
<?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
<?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
<?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
<?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

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

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

b. Working with Arrays

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

c. Working with Files (Reading/Writing)

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

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

```php
  // 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
  <?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
  )";
```

```php
    if ($conn->query($sql) === TRUE) {
       echo "Table Student created successfully.<br>";
    } else {
       echo "Error creating table: " . $conn->error;
    }
    // Insert 3 records
    $sql = "INSERT INTO Student (sname, percentage) VALUES
          ('Sam', 45.5),
          ('Raju', 67.8),
          ('Ram', 28.4)";
    if ($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
<?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);
}
```

```php
// 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();
?>
```

b. **Delete rows from a table**

```php
<?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
<?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>
```

9. Write PHP scripts for
a. Storing and Retrieving Cookies

setCookie.php
```php
<?php
// Set a cookie named 'user' with a value of 'Gufran Sir' that expires in 7 days
$cookie_name = "user";
$cookie_value = "Gufran Sir";
$expiry_time = time() + (7 * 24 * 60 * 60); // 7 days from now

setcookie($cookie_name, $cookie_value, $expiry_time, "/"); // '/' means available across the entire website

echo "Cookie named 'user' has been set!";
?>
```

getCookie.php
```php
<?php
// Check if the 'user' cookie is set
if(isset($_COOKIE["user"])) {
    echo "User Cookie Value: " . $_COOKIE["user"];
} else {
    echo "User cookie is not set.";
}
?>
```

b. Storing and Retrieving Sessions

setSession.php

```php
<?php
session_start(); // Start the session
// Store session data
$_SESSION["user"] = "Gufran Sir";
echo "Session variable 'user' has been set!";
?>
```

getSession.php

```php
<?php
session_start(); // Start the session

// Check if the session variable 'user' exists
if(isset($_SESSION["user"])) {
    echo "User Session Value: " . $_SESSION["user"];
} else {
    echo "User session is not set.";
}
?>
```

10. Perform the following using Bootstrap:

a. Create a responsive layout using the Bootstrap grid system

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bootstrap Layout</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body class="d-flex flex-column min-vh-100">
    <div class="container flex-grow-1">
        <header class="bg-primary text-white text-center py-3 mb-3">
            <h1>My Website</h1>
        </header>
        <div class="row">
            <div class="col-md-8 p-3 bg-light">
                <h2>Main Content</h2>
                <p>Responsive layout with Bootstrap grid.</p>
            </div>
            <div class="col-md-4 p-3 bg-secondary text-white">
                <h2>Sidebar</h2>
                <p>Additional information goes here.</p>
            </div>
        </div>
```

```
    </div>
    <footer class="bg-dark text-white text-center py-2 mt-auto">
      <p>&copy; 2025 My Website</p>
    </footer>
  </body>
</html>
```

b. Create a simple Bootstrap navbar with dropdown menus

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Navbar with Dropdown</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css"
rel="stylesheet">
</head>
<body>
  <nav class="navbar navbar-expand-lg bg-primary navbar-dark">
    <div class="container">
      <a class="navbar-brand" href="#">MyWebsite</a>
      <button class="navbar-toggler" data-bs-toggle="collapse" data-bs-target="#navbarNav">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="navbarNav">
        <ul class="navbar-nav ms-auto">
          <li class="nav-item"><a class="nav-link active" href="#">Home</a></li>
          <li class="nav-item"><a class="nav-link" href="#">About</a></li>
          <li class="nav-item dropdown">
            <a class="nav-link dropdown-toggle" href="#" data-bs-
toggle="dropdown">Services</a>
            <ul class="dropdown-menu">
              <li><a class="dropdown-item" href="#">Web Design</a></li>
              <li><a class="dropdown-item" href="#">SEO</a></li>
              <li><a class="dropdown-item" href="#">Marketing</a></li>
            </ul>
          </li>
          <li class="nav-item"><a class="nav-link" href="#">Contact</a></li>
        </ul>
      </div>
    </div>
  </nav>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>

</body>
</html>
```

c. Create a basic Bootstrap form with validation

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap Form with Validation</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet">
</head>
<body>
  <div class="container mt-5">
    <h2>Contact Form</h2>
    <form id="contactForm" class="needs-validation" novalidate>
      <div class="mb-3">
        <label for="name" class="form-label">Full Name</label>
        <input type="text" class="form-control" id="name" required>
        <div class="invalid-feedback">Please enter your name.</div>
      </div>

      <div class="mb-3">
        <label for="email" class="form-label">Email address</label>
        <input type="email" class="form-control" id="email" required>
        <div class="invalid-feedback">Please provide a valid email address.</div>
      </div>
      <div class="mb-3">
        <label for="message" class="form-label">Message</label>
        <textarea class="form-control" id="message" rows="4" required></textarea>
        <div class="invalid-feedback">Please enter a message.</div>
      </div>
      <button type="submit" class="btn btn-primary">Submit</button>
    </form>
  </div>

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.min.js"></script>
  <script>
    document.getElementById('contactForm').addEventListener('submit', function (event) {
      if (!this.checkValidity()) {
        event.preventDefault();
        event.stopPropagation();
      }
      this.classList.add('was-validated');
    });
  </script>
</body>
</html>
```