**Practical-2**

**Aim :- Create interactive webpages using Javascript**.

**Hardware Requirement :-**

1. Intel I5 System.

1. Good Graphics Card.
2. Internet Hardware ex-Wi-Fi, Wired Connection etc.

**Software Requirement :-**

1. Browser ex- Chrome, opera, firefox, etc,.
2. Application to code Html Commands ex- VS Code, Notpad, etc,.
3. Active Internet Connection with any Browser

**Knowledge Requirement :-**

1. Basic knowledge of Html tags.
2. Basic knowledge of JavaScript.

1. **Write a program to print hello world Code :**

<html>

<head>

<script type="text/javascript"> document.write("Hello World");

</script>

</head>

<body></body>

</html>

**Output :**

A screenshot of a video game

Description automatically generated

1. **Write a JavaScript to demonstrate user define function**

**:**

<html>

<head>

<script type="text/javascript">

function Hello(){

document.write("Hello World");

}

</script>

</head>

<body>

<input type="button" onclick="Hello()">

</body>

</html> **Output :**

A screenshot of a computer

Description automatically generated

**3. Write a JavaScript to demonstrate Operators**

**Code :**

<html>

<body>

<script type="text/javascript"> a=5; b=10; c=a+b;

document.write("a+b="+c);

</script>

</body> </html>

**Output :**

A screen shot of a computer

Description automatically generated

# 4. Write a JavaScript to demonstrate loop statement

<html>

<body>

<script type="text/javascript">

i=0; do{

i++;

document.write(i+" ");

} while (i<10)

</script>

<br><br>

<script> i=0; while(i<10){ i++; document.write(i+" ")

}

</script>

<br><br>

<script> for(i=1;i<10;i++) document.write(i+" ")

</script>

</body>

</html>

**Output :**

A screenshot of a computer

Description automatically generated

# 5. Write a JavaScript to display alert pop up

**:**

<html>

<head>

<title>Alert Box Example</title>

</head>

<body>

<button onclick="showAlert()">Click Me</button>

<script> function showAlert() { alert("This is an alert box!");

}

</script>

</body>

</html> **Output :**

A screenshot of a computer

Description automatically generated

**6. Write a JavaScript to get a prompt from user**

**:**

<html>

<head>

<title>Prompt Box Example</title>

</head> <body>

<button onclick="showPrompt()">Click Me</button>

<script> function showPrompt() { var userInput

= prompt("Please enter your name:"); if

(userInput !== null) { alert("Hello, " + userInput + "!");

} else { alert("You canceled the prompt.");

}

}

</script>

</body>

</html> **Output :**

A screenshot of a login form

Description automatically generated

A white rectangular object with a black border

Description automatically generated

**7. Write a JavaScript to display confirm pop up**

**:**

<html>

<head>

<title>Confirm Box Example</title>

</head>

<body>

<button onclick="showConfirm()">Click Me</button>

<script> function showConfirm() { var result = confirm("Are you sure you want to delete this item?");

if (result) {

alert("Item deleted.");

} else {

alert("Deletion canceled.");

}

}

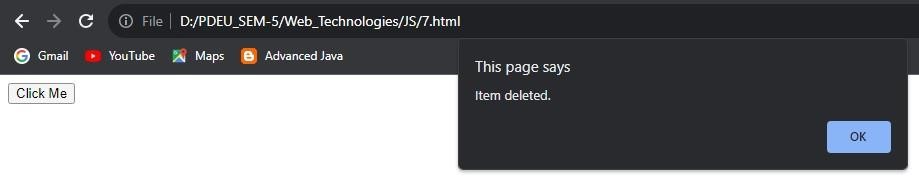
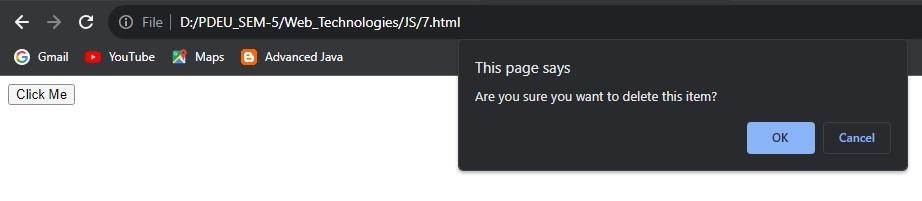
</script>

</body>

</html>

**Output**

**:**



# 8. Write a JavaScript to validate form input

**:**

<!DOCTYPE html>

<html>

<head>

<title>Basic Form Validation</title>

</head>

<body>

<h2>Basic Form Validation</h2>

<form id="myForm">

<label for="name">Name:</label>

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

<label for="email">Email:</label>

<input type="email" id="email" name="email"><br><br>

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

</form>

<p id="error" style="color: red;"></p>

<script> var form = document.getElementById('myForm'); form.addEventListener('submit', function (event) { event.preventDefault();

var name = document.getElementById('name').value;

var email = document.getElementById('email').value;

if (name === '' || email === '') {

document.getElementById('error').textContent = 'Both fields are required!

} else {

document.getElementById('error').textContent = ''; alert('Form submitted successfully!'); form.reset();

}

});

</script>

</body>

</html> **Output :**

A screenshot of a computer

Description automatically generated

**Conclusion :**

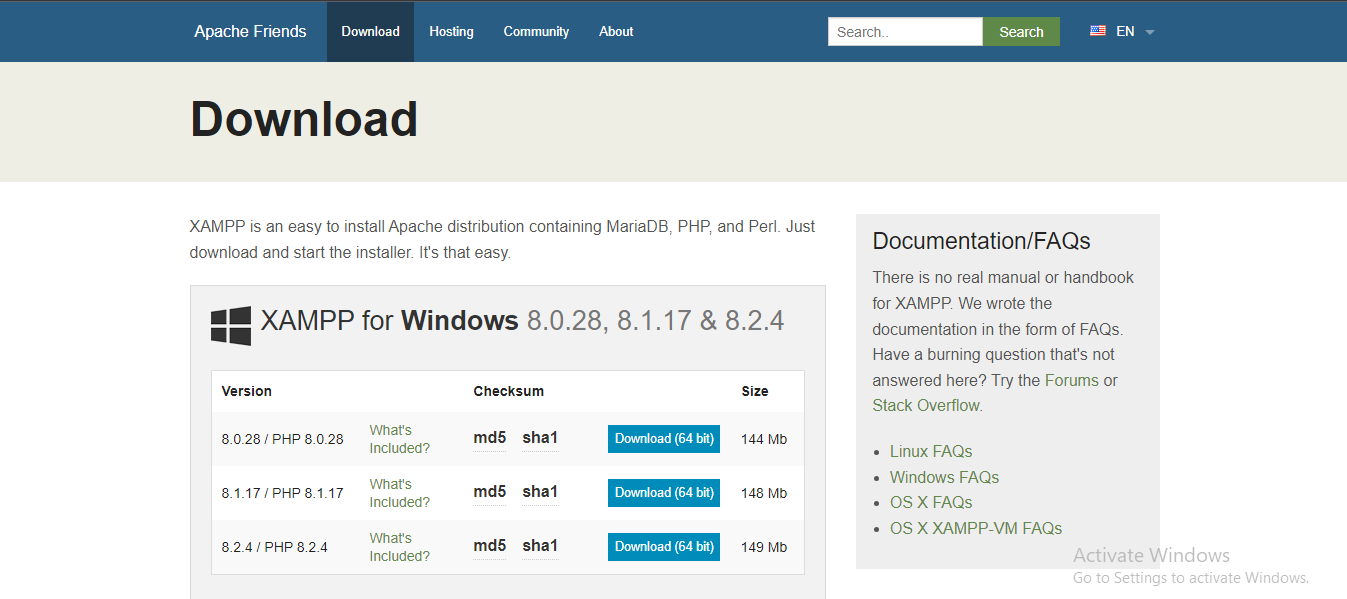
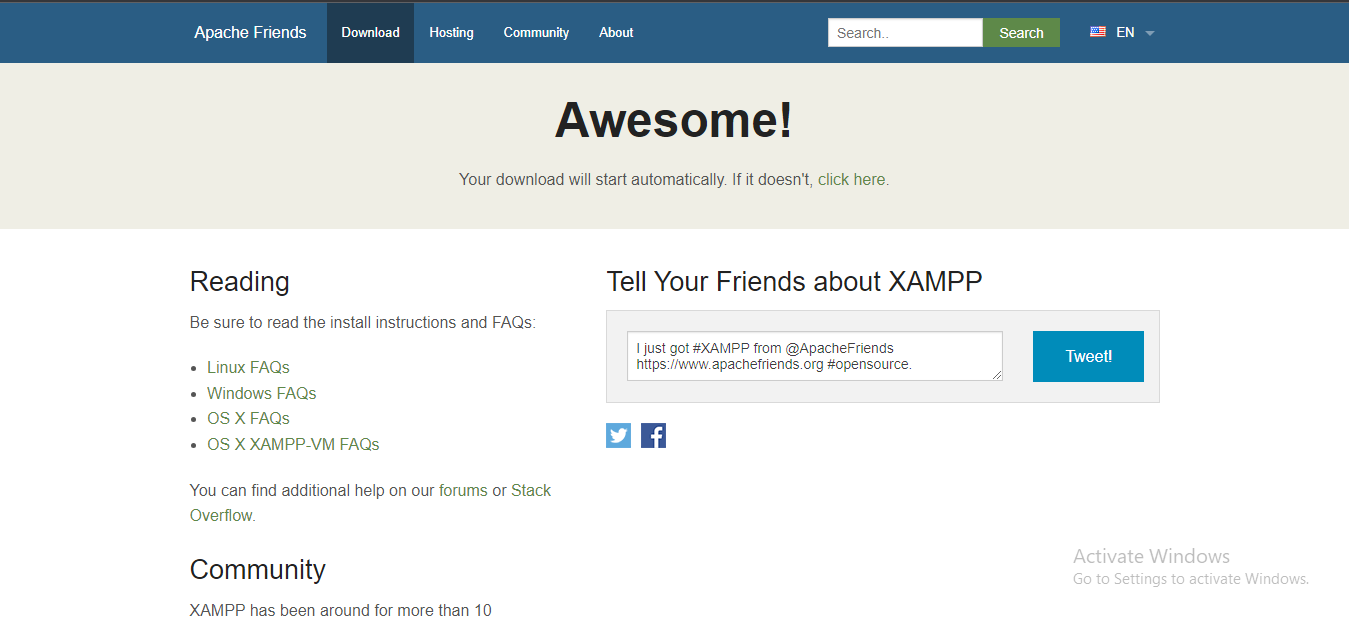
* In conclusion, acquiring a fundamental understanding of JavaScript provides a solid foundation for web development and opens the door to creating dynamic and interactive web applications.
* Further exploration and practice in this versatile language will undoubtedly enhance one's coding skills and career opportunities.

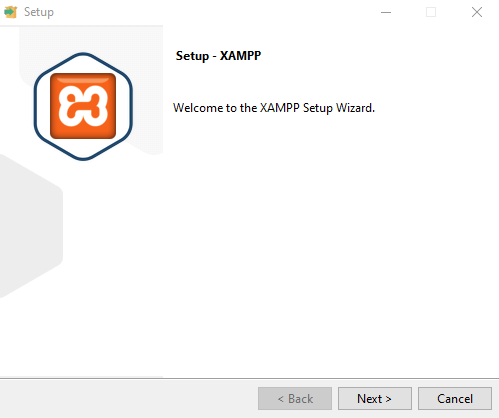
**References :**

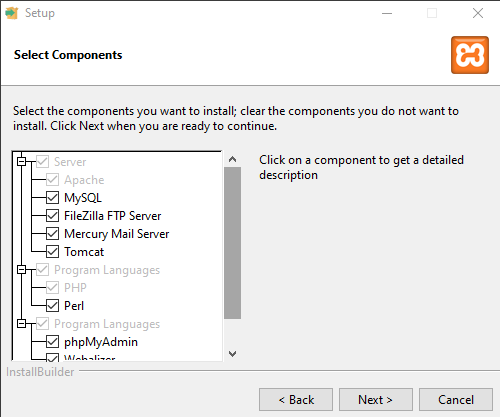
<https://www.w3schools.com/js>[/](https://www.w3schools.com/js/)

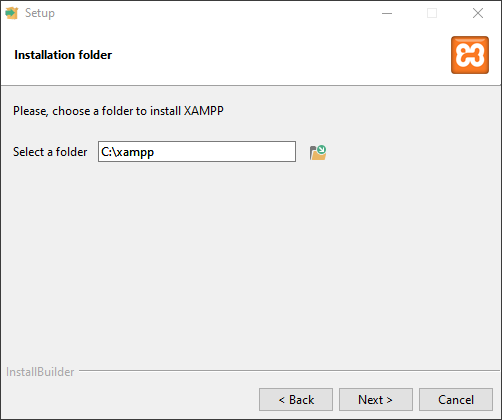
[https://www.javatpoint.com/javascripttutorial](https://www.javatpoint.com/javascript-tutorial)

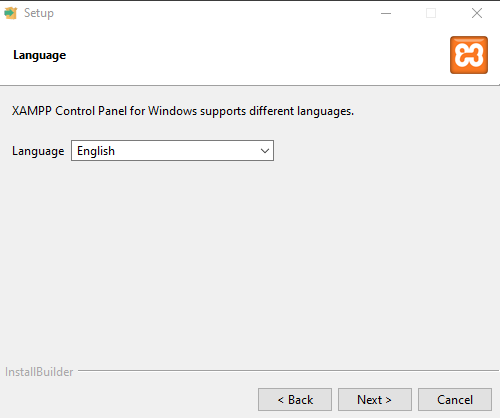
**Practical-3**

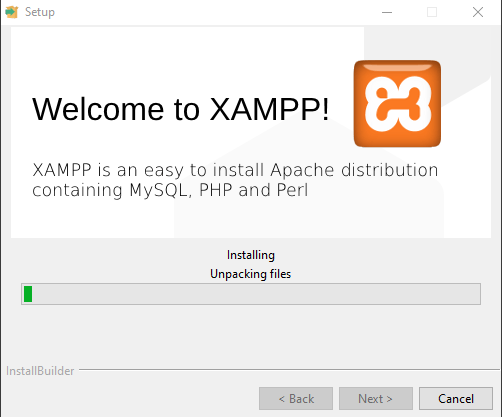


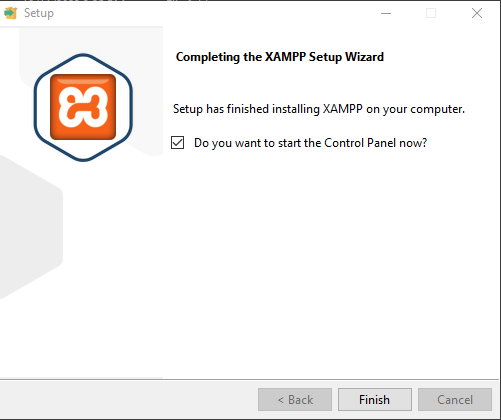


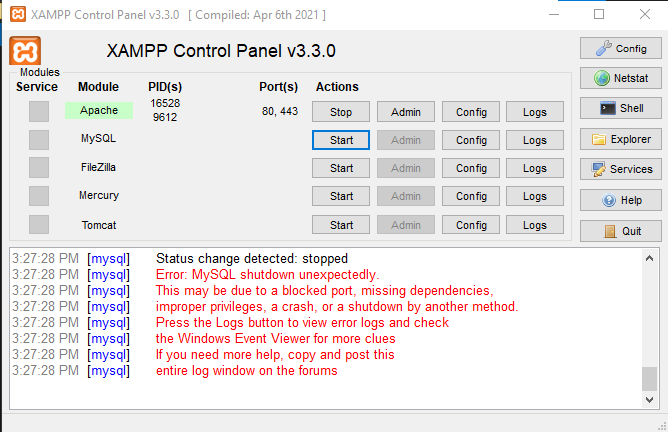












**Practical-4**

**Aim :- IMPLEMENT the server side scripting using php language**

**Hardware Requirement :-**

1. **Computer:** A desktop or laptop computer with sufficient processing power and memory to run web applications. XAMPP is not resource-intensive, so most modern computers should suffice.
2. **Storage Space:** Adequate storage space for your web development projects and the XAMPP software. A few gigabytes of free space should be more than enough.
3. **RAM:** At least 2GB of RAM for smooth operation. More RAM may be required if you plan to run resource-intensive applications.

**Software Requirement :-**

1. **Operating System:** XAMPP is compatible with Windows, macOS, and Linux. Ensure your system meets the following OS-specific requirements:

- Windows: XP, Vista, 7, 8, 10

- macOS: 10.6 or later

- Linux: Any modern distribution with a 32- or 64-bit architecture.

2. **Web Browser:** You'll need a web browser for testing your web applications. Popular choices include Google Chrome, Mozilla Firefox, and Microsoft Edge.

3. **XAMPP Software:** Download and install the XAMPP software from the official website (https://www.apachefriends.org/). Make sure to download the version that matches your operating system.

4. **Text Editor or IDE:** A text editor or integrated development environment (IDE) for writing and editing your web application code. Popular options include Visual Studio Code, Sublime Text, and PHP Storm.

5. **Database Management Tool:** If your web application uses databases, you may need a database management tool such as phpMyAdmin (included in XAMPP) or MySQL Workbench.

6. **Optional: Version Control System:** Consider using a version control system (e.g., Git) for tracking changes in your code.

7. **Optional:** Content Management System (CMS):\*\* If you plan to use a CMS like WordPress, Joomla, or Drupal, make sure to meet their specific software requirements.

**Knowledge Requirement :-**

**1. Basic Programming Concepts:**

- Variables, data types, and operators.

- Control structures (if statements, loops).

- Functions and procedures.

**2. PHP Syntax:**

- Familiarity with PHP syntax, including variables, arrays, loops, and functions.

**3. Web Development Basics:**

- Understanding of HTML and CSS.

- Knowledge of how web servers work.

**4. Server-Side Scripting:**

- Understanding of the difference between server-side and client-side scripting.

**5. PHP Data Types:**

- Scalars (int, float, string, boolean).

- Composite data types (arrays, objects).

- Special data types (NULL).

**6. Functions and Control Structures:**

- How to define and call functions.

- Conditional statements (if, else, switch).

- Looping (for, while, foreach).

**7. Super Globals:**

- Knowledge of PHP's superglobal arrays like `$\_GET`, `$\_POST`, `$\_SESSION`, and `$\_COOKIE`.

**8. Error Handling:**

- Understanding of error reporting and handling mechanisms.

**9. Working with Databases:**

- Connecting to databases using PHP (e.g., MySQL, PostgreSQL, SQLite).

- Executing SQL queries safely.

**10. Form Handling:**

- Handling form submissions and user input validation.

**11. Sessions and Cookies:**

- Working with sessions and cookies for user state management.

**12. File Handling:**

- Reading and writing files, and working with directories.

**13. Security Best Practices:**

- Awareness of common security vulnerabilities (e.g., SQL injection, XSS) and how to mitigate them.

**14. Object-Oriented Programming (OOP):**

- Basics of OOP in PHP, including classes, objects, inheritance, and encapsulation.

**15. MVC (Model-View-Controller) Architecture:**

- Understanding the concept of separating the application into models, views, and controllers.

**16. API Integration:**

- How to work with RESTful and other types of APIs in PHP.

**17. Frameworks and Libraries:**

- Familiarity with popular PHP frameworks (e.g., Laravel, Symfony, CodeIgniter) and libraries.

**18. Composer:**

- Dependency management using Composer, a PHP package manager.

**19. Debugging and Profiling:**

- Using debugging tools and techniques to find and fix issues in your code.

**20. Version Control:**

- Using version control systems like Git to manage your PHP projects.

**21. Server Configuration:**

- Knowledge of server configurations, such as PHP.ini settings.

**22. Caching:**

- Implementing caching mechanisms to improve application performance.

**23. Web Security:**

- Knowledge of best practices for securing PHP applications and servers.

**24. Regular Expressions:**

- Understanding and using regular expressions for pattern matching.

**25. Web Application Deployment:**

- Deploying PHP applications on web servers.

**26. Performance Optimization:**

- Techniques for optimizing PHP code and improving application performance.

**27. API Documentation:**

- How to create and document APIs in PHP.

**28. Testing and Test-Driven Development (TDD):**

- Strategies for testing PHP code and implementing TDD practices.

**29. Continuous Integration/Continuous Deployment (CI/CD):**

- Integrating PHP projects into CI/CD pipelines for automated testing and deployment.

**Theory:**

PHP, which stands for "Hypertext Preprocessor," is a popular server-side scripting language primarily used for web development. It was created by Rasmus Lerdorf in 1994 and has since evolved into one of the most widely used languages for building dynamic web applications. Here are some key points about PHP:

**1. Server-Side Scripting:** PHP is executed on the web server, which means it processes code on the server side before sending the resulting HTML to the client's web browser. This allows for the generation of dynamic web content.

**2. Open Source:** PHP is an open-source language, which means it is freely available for anyone to use, modify, and distribute. This has contributed to its widespread adoption and the development of a vast community and ecosystem.

**3. Embeddable in HTML:** PHP code is typically embedded directly into HTML, allowing developers to mix dynamic and static content seamlessly. PHP code is enclosed in `<?php ... ?>` tags.

**4. Cross-Platform:** PHP is platform-independent, meaning it can run on various operating systems, including Windows, Linux, macOS, and more.

**5. Extensive Community and Ecosystem:** PHP has a large and active community of developers, which has led to the creation of numerous frameworks, libraries, and tools for web development.

**6.Support for Databases:** PHP supports various databases, such as MySQL, PostgreSQL, SQLite, and more, making it a popular choice for database-driven web applications.

**7. Server Compatibility:** PHP works with most web servers, with Apache being one of the most commonly used in combination with PHP.

**8. Object-Oriented Programming (OOP):** PHP has full support for OOP, allowing developers to write modular and maintainable code using classes and objects.

**9. Popular Frameworks:** PHP has several popular frameworks, including Laravel, Symfony, and CodeIgniter, which provide pre-built structures and libraries for building web applications.

**10. Security Concerns:** Like any web development language, PHP has security considerations, and developers must be aware of best practices to prevent common vulnerabilities like SQL injection, Cross-Site Scripting (XSS), and more.

**11. Versatile:** PHP can be used for various web-related tasks, including building websites, web applications, content management systems (CMS), e-commerce platforms, and APIs.

**12. Community Support:** PHP's extensive documentation and community forums make it relatively easy for developers to find help and resources when they encounter problems or have questions.

**13. Continuous Development:** PHP continues to evolve with new versions and features. As of my last knowledge update in September 2021, PHP 8 was the latest major release, bringing significant improvements to the language.

1. **Basic code for Introduction of php**

**Code :**

<!DOCTYPE html>

<html>

<body>

<?php

echo "My first PHP script!";

?>

</body>

</html>

A screenshot of a computer

Description automatically generated**Output :**

1. **Variable Declaration**

**Code :**

<?php

$str="hello Kavit";

$x=200;

$y=44.6;

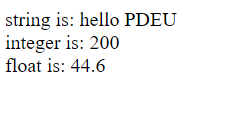
echo "string is: $str <br/>";

echo "integer is: $x <br/>";

echo "float is: $y <br/>";

?>

**Output :**



1. **Operator in PHP**

**Code :**

<?php

$x=25;

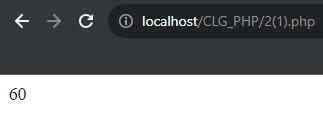
$y=35;

$z=$x+$y;

echo $z;

?>

**Output :**

****

1. **Check the given number is even or odd(control statement)**

**Code :**

<?php

$num=12;

if($num%2==0){

echo "$num is even number";

}else{

echo "$num is odd number";

}

?>

**Output :**

A screenshot of a computer

Description automatically generated

1. **Write a program to Demonstrate the Array**

**Code :**

<?php

$season=array("summer","winter","spring","autumn");

echo "Season are: $season[0], $season[1], $season[2] and $season[3]";

?>

**Output :**

A screenshot of a computer

Description automatically generated

1. **Write a program to demonstrate the Loops**

**Code :**

<?php

for($n=1;$n<=10;$n++){

echo "$n<br/>";

}

?>

**Output :**

A screenshot of a computer

Description automatically generated

1. **Write a program to Demonstrate the Associative Array**

**Code :**

<?php

$salary=array("Aayush"=>"350000","Kathan"=>"450000","Kirtan"=>"200000");

echo "Aayush salary: ".$salary["Aayush"]."<br/>";

echo "Kathan salary: ".$salary["Kathan"]."<br/>";

echo "Kirtan salary: ".$salary["Kirtan"]."<br/>";

?>

**Output :**

A screenshot of a computer

Description automatically generated

1. **Write a program to Demonstrate the Form handling in PHP**

**Code :**

**(get method)**

<form action="form1.php" method="get">

Name: <input type="text" name="name"/>

<input type="submit" value="visit"/>

</form>

<?php

$name=$\_GET["name"];

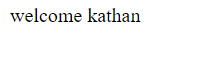
echo "Welcome, $name";

?>

**Output :**

A white rectangular object with a blue stripe

Description automatically generated



**(post method)**

**Code :**

<form action="form2.php" method="post">

<table>

<tr><td>Name:</td><td> <input type="text" name="name"/></td></tr>

<tr><td>Password:</td><td> <input type="password" name="password"/></td></tr>

<tr><td colspan="2"><input type="submit" value="login"/> </td></tr>

</table>

</form>

<?php

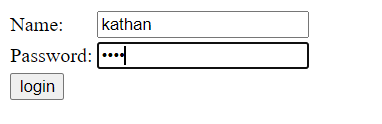
$name=$\_POST["name"];

$password=$\_POST["password"];

echo "Welcome: $name, your password is: $password";

?>

**Output :**





1. **Write a program in php for from validation**

**Code :**

<html>

<head>

</head>

<body>

<?php

$name = $email = $gender = $comment = $website = "";

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

$name = test\_input($\_POST["name"]);

$email = test\_input($\_POST["email"]);

$website = test\_input($\_POST["website"]);

$comment = test\_input($\_POST["comment"]);

$gender = test\_input($\_POST["gender"]);

}

function test\_input($data) {

$data = trim($data);

$data = stripslashes($data);

$data = htmlspecialchars($data);

return $data;

}

?>

<h2>PHP Form Validation Example</h2>

<form method="post" action="<?php echo

htmlspecialchars($\_SERVER["PHP\_SELF"]);?>">

Name: <input type="text" name="name">

<br><br>

E-mail: <input type="text" name="email">

<br><br>

Website: <input type="text" name="website">

<br><br>

Comment: <textarea name="comment" rows="5" cols="40"></textarea>

<br><br>

Gender:

<input type="radio" name="gender" value="female">Female

<input type="radio" name="gender" value="male">Male

<input type="radio" name="gender" value="other">Other

<br><br>

<input type="submit" name="submit" value="Submit">

</form>

<?php

echo "<h2>Your Input:</h2>";

echo $name;

echo "<br>";

echo $email;

echo "<br>";

echo $website;

echo "<br>";

echo $comment;

echo "<br>";

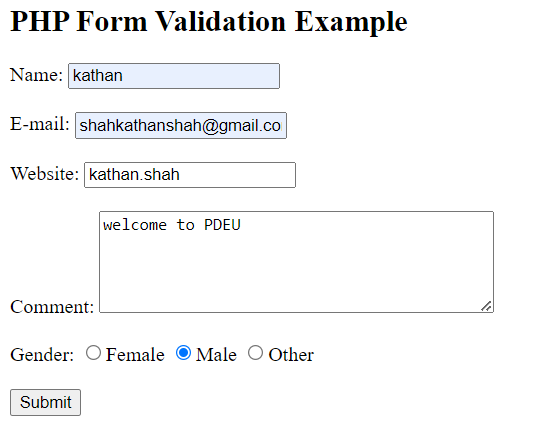
echo $gender;

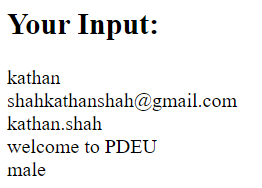
?>

</body>

</html>

**Output :**





1. **Write a program to perform Require validation on form**

**Code :**

<html>

<head>

<style>

.error {color: #FF0000;}

</style>

</head>

<body>

<?php

// define variables and set to empty values

$nameErr = $emailErr = $genderErr = $websiteErr = "";

$name = $email = $gender = $comment = $website = "";

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

if (empty($\_POST["name"])) {

$nameErr = "Name is required";

} else {

$name = test\_input($\_POST["name"]);

}

if (empty($\_POST["email"])) {

$emailErr = "Email is required";

} else {

$email = test\_input($\_POST["email"]);

}

if (empty($\_POST["website"])) {

$website = "";

} else {

$website = test\_input($\_POST["website"]);

}

if (empty($\_POST["comment"])) {

$comment = "";

} else {

$comment = test\_input($\_POST["comment"]);

}

if (empty($\_POST["gender"])) {

$genderErr = "Gender is required";

} else {

$gender = test\_input($\_POST["gender"]);}}

function test\_input($data) {

$data = trim($data);

$data = stripslashes($data);

$data = htmlspecialchars($data);

return $data;

}

?>

<h2>PHP Form Validation Example</h2>

<p><span class="error">\* required field</span></p>

<form method="post" action="<?php echo

htmlspecialchars($\_SERVER["PHP\_SELF"]);?>">

Name: <input type="text" name="name">

<span class="error">\* <?php echo $nameErr;?></span>

<br><br>

E-mail: <input type="text" name="email">

<span class="error">\* <?php echo $emailErr;?></span>

<br><br>

Website: <input type="text" name="website">

<span class="error"><?php echo $websiteErr;?></span>

<br><br>

Comment: <textarea name="comment" rows="5" cols="40"></textarea>

<br><br>

Gender:

<input type="radio" name="gender" value="female">Female

<input type="radio" name="gender" value="male">Male

<input type="radio" name="gender" value="other">Other

<span class="error">\* <?php echo $genderErr;?></span>

<br><br>

<input type="submit" name="submit" value="Submit">

</form>

<?php

echo "<h2>Your Input:</h2>";

echo $name;

echo "<br>";

echo $email;

echo "<br>";

echo $website;

echo "<br>";

echo $comment;

echo "<br>";

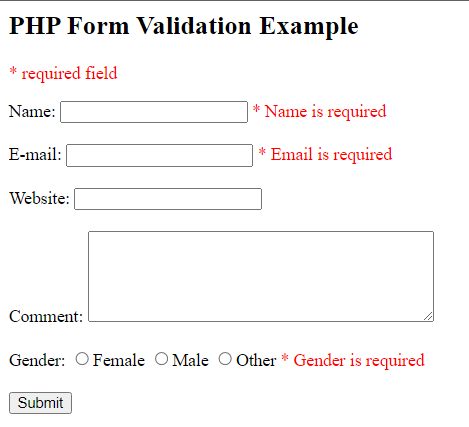
echo $gender;

?>

</body>

</html>

**Output :**



1. **Write a code to create cookie in PHP**

**Code :**

<?php

setcookie("user", "Kavit");

?>

<html>

<body>

<?php

if(!isset($\_COOKIE["user"])) {

echo "Sorry, cookie is not found!";

} else {

echo "<br/>Cookie Value: " . $\_COOKIE["user"];

}

?>

</body>

</html>

**Output :**



1. **Write a code to delete a cookie In PHP**

**Code :**

<?php

setcookie ("c1", "", time() - 3600);

echo "Cookie will be deleted in"

?>

**Output :**



1. **Write a program to demonstrate th session in php**

**Code :**

<?php

session\_start();

?>

<html>

<body>

<?php

$\_SESSION["user"] = "Kavit";

echo "Session information are set successfully.<br/>";

?>

<a href="session2.php">Visit next page</a>

</body>

</html>

<?php

session\_start();

?>

<html>

<body>

<?php

echo "User is: ".$\_SESSION["user"];

?>

</body>

</html>

**Output :**



1. **Write a program to counter in php**

**Code :**

<?php

session\_start();

if (!isset($\_SESSION['counter'])) {

$\_SESSION['counter'] = 1;

} else {

$\_SESSION['counter']++;

}

echo ("Page Views: ".$\_SESSION['counter']);

?>

**Output :**



1. **Write a php code to open a file**

**Code :**

<?php

$filename = "C:\\xampp\\htdocs\\CLG\_PHP\\file.txt";

$handle = fopen($filename, "r");

$contents = fread($handle, filesize($filename));

echo $contents;

fclose($handle);

?>

**Output :**



1. **Write a PHP code to write a file**

**Code :**

<?php

$fp = fopen("C:\\xampp\htdocs\\CLG\_PHP\\file.txt", 'w');

fwrite($fp, 'hello ');

fwrite($fp, 'php file');

fclose($fp);

echo "File written successfully<br>";

echo "File Closed Successfully";

?>

**Output :**



1. **Write a php code to Read a file**

**Code :**

<?php

$filename = "C:\\xampp\\htdocs\\CLG\_PHP\\file.txt";

$handle = fopen($filename, "r");

$contents = fread($handle, filesize($filename));

echo $contents;

fclose($handle);

?>

**Output :**

****

1. **Write the a PHP code to Delete the File**

**Code :**

<?php

unlink("C:\\xampp\\htdocs\\CLG\_PHP\\file.txt");

echo "File deleted successfully"; ?>

**Output :**

****

1. **Write a PHP code to Upload a file**

**Code :**

<form action="uploader.php" method="post" enctype="multipart/form-data">

Select File:

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

<input type="submit" value="Upload Image" name="submit"/>

</form>

<?php

$target\_path = "e:/";

$target\_path = $target\_path.basename( $\_FILES['fileToUpload']['name']);

if(move\_uploaded\_file($\_FILES['fileToUpload']['tmp\_name'], $target\_path)) {

echo "File uploaded successfully!";

} else{

echo "Sorry, file not uploaded, please try again!";

}

?>

**Output :**

A black text on a white background

Description automatically generated



**Conclusion:**

* In conclusion, PHP is a mature, versatile, and open-source scripting language commonly used for web development. It enjoys a large community, offers various frameworks for efficient development, and has improved in terms of performance and security over the years. It remains a popular choice for building web applications of all sizes.

**References :**

[https://www.w3schools.com](https://www.w3schools.com/)

[https://www.javatpoint.com](https://www.javatpoint.com/)

<https://www.apachefriends.org/download.html>

**Practical: 5**

**Aim:** Create a web page that retrieves and displays info from the XML file

**Requirement:** Laptop /Computer

**Software Requirement:** -

- Windows 10/11 / Browser (Chrome, Brave, Firefox) / Winget

- PHP / XAMPP / VS code / Notepad

**Knowledge Requirement:** HTML, CSS, PHP

## Ajax2.html

<!DOCTYPE html>

<html>

<style> table, th,

td {

border: 1px solid black;

border-collapse: collapse;

}

th,

td {

padding: 5px;

}

</style>

<body>

<button type="button" onclick="loadXMLDoc()">Get my CD collection</button>

<br /><br />

<table id="demo"></table>

<script>

function loadXMLDoc() {

var xmlhttp = new XMLHttpRequest(); xmlhttp.onreadystatechange = function () {

if (this.readyState == 4 && this.status == 200) { myFunction(this);

}

};

xmlhttp.open("GET", "cd\_catalog.xml", true);

xmlhttp.send();

}

function myFunction(xml) { var i;

var xmlDoc = xml.responseXML;

var table = "<tr><th>Artist</th><th>Title</th></tr>"; var x = xmlDoc.getElementsByTagName("CD");

for (i = 0; i < x.length; i++) { table +=

"<tr><td>" + x[i].getElementsByTagName("ARTIST")[0].childNodes[0].nodeValue + "</td><td>" + x[i].getElementsByTagName("TITLE")[0].childNodes[0].nodeValue + "</td></tr>";

}

document.getElementById("demo").innerHTML = table;

}

</script>

</body>

</html>

## Cd\_catalog.xml

<?xml version="1.0" encoding="UTF-8"?>

<CATALOG>

<CD>

<ARTIST>The Beatles</ARTIST>

<TITLE>Abbey Road</TITLE>

</CD>

<CD>

<ARTIST>Pink Floyd</ARTIST>

<TITLE>The Dark Side of the Moon</TITLE>

</CD>

<CD>

<ARTIST>Michael Jackson</ARTIST>

<TITLE>Thriller</TITLE>

</CD>

<CD>

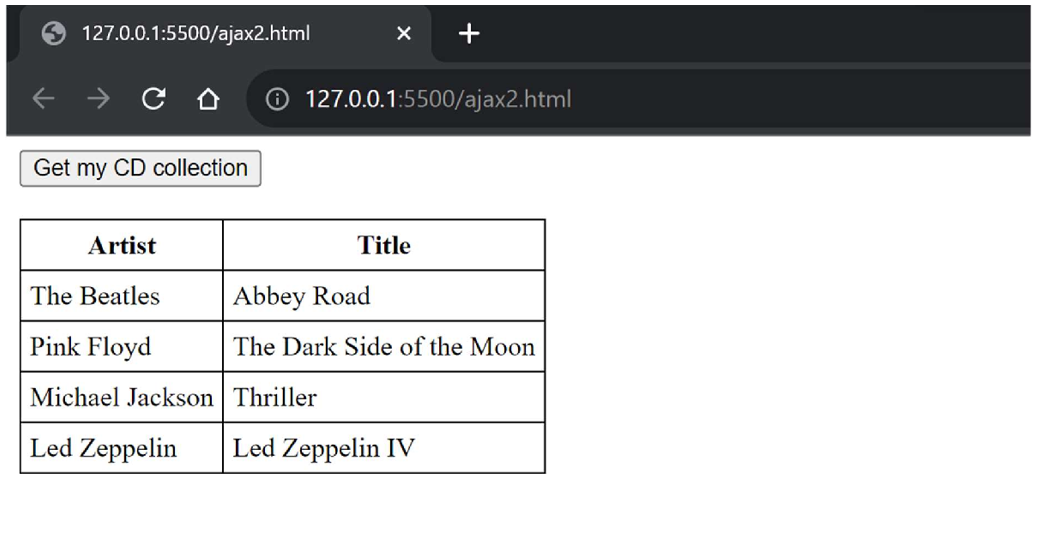
<ARTIST>Led Zeppelin</ARTIST>

<TITLE>Led Zeppelin IV</TITLE>

</CD>

</CATALOG>

**OUTPUT:**

****

**Practical: 6**

## **Aim:** Create a web page that retrieves and displays info from the Json file

**Requirement:** Laptop /Computer

**Software Requirement:** -

## - Windows 10/11 / Browser (Chrome, Brave, Firefox) / Winget

- VS code / Notepad

**Knowledge Requirement:** HTML, CSS, JS

### Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8" />

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

<title>Music Records</title>

<style> table, th,

td {

border: 1px solid black; border-collapse: collapse;

}

th,

td {

padding: 5px; text-align: left;

}

th {

background-color: #f2f2f2;

}

</style>

</head>

<body>

<button type="button" onclick="loadMusicData()">Show Data</button>

<br /><br />

<table id="musicTable">

<thead>

<tr>

<th>Artist</th>

<th>Title</th>

<th>Genre</th>

<th>Album Year</th>

</tr>

</thead>

<tbody>

<!-- Data will be inserted here -->

</tbody>

</table>

<script>

function loadMusicData() {

var xhr = new XMLHttpRequest(); xhr.onreadystatechange = function () {

if (this.readyState == 4 && this.status == 200) { var musicData = JSON.parse(this.responseText); populateTable(musicData);

}

};

xhr.open("GET", "music\_records.json", true); xhr.send();

}

function populateTable(data) { var table = document

.getElementById("musicTable")

.getElementsByTagName("tbody")[0]; table.innerHTML = "";

data.forEach(function (record) { var row = table.insertRow();

row.insertCell(0).textContent = record.artist; row.insertCell(1).textContent = record.title; row.insertCell(2).textContent = record.genre; row.insertCell(3).textContent = record.albumYear;

});

}

</script>

</body>

</html>

### music\_records.json

[

{

"artist": "The Beatles", "title": "Abbey Road", "genre": "Rock", "albumYear": "1969"

},

{

"artist": "Michael Jackson", "title": "Thriller",

"genre": "Pop", "albumYear": "1982"

},

{

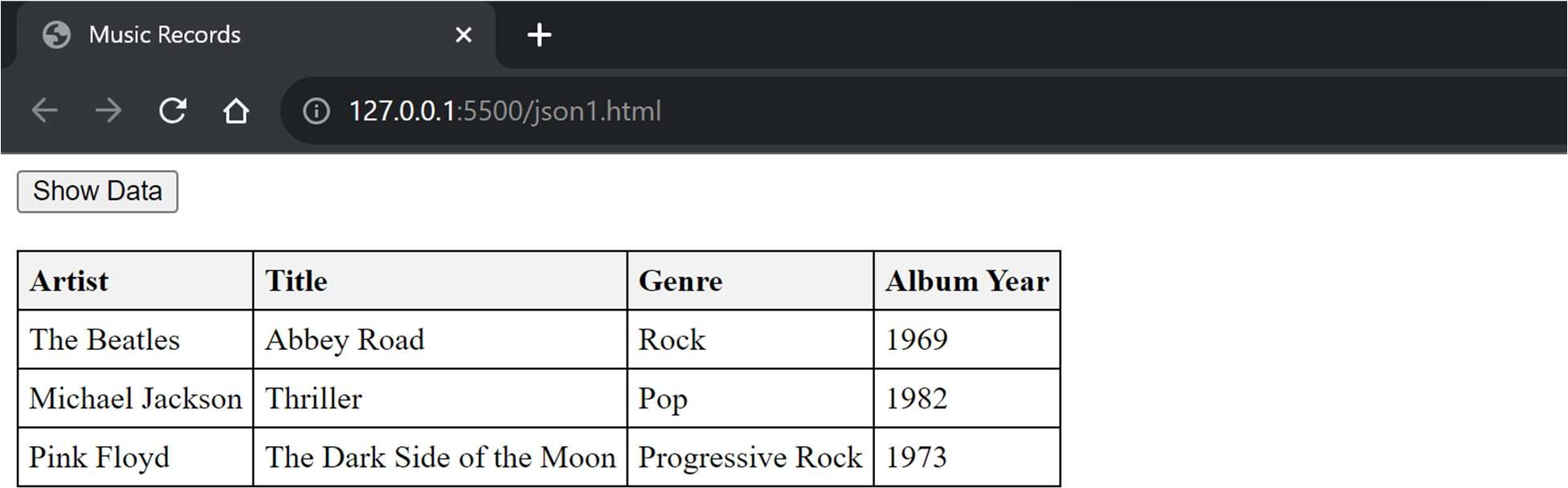
"artist": "Pink Floyd",

"title": "The Dark Side of the Moon", "genre": "Progressive Rock", "albumYear": "1973"

}

]

**OUTPUT:**



**Practical: 7**

## **Aim:** Implement the web applications using PHP and add the AJAX feature into it.

**Requirement:** Laptop /Computer

**Software Requirement:** -

## - Windows 10/11 / Browser (Chrome, Brave, Firefox) / Winget

- PHP / XAMPP / VS code / Notepad

**Knowledge Requirement:** HTML, CSS, PHP, XML

### myxml.php

<?php

// Load and parse the XML file

$xml = simplexml\_load\_file('data.xml');

// Check if the XML was loaded successfully if ($xml === false) {

echo '<p>Failed to load XML data.</p>';

} else {

// Output the information

echo '<h1>Book Information</h1>'; foreach ($xml->book as $book) {

echo '<div class="book">';

echo '<h2>Title: ' . htmlspecialchars($book->title) . '</h2>'; echo '<p>Author: ' . htmlspecialchars($book->author) . '</p>'; echo '<p>Genre: ' . htmlspecialchars($book->genre) . '</p>'; echo '<p>Published: ' . htmlspecialchars($book->published) .

'</p>';

}

}

?>

echo '</div>';

### data.xml

<?xml version="1.0" encoding="UTF-8"?>

<library>

<book>

<title>1984</title>

<author>George Orwell</author>

<genre>Dystopian</genre>

<published>1949</published>

</book>

<book>

<title>Brave New World</title>

<author>Aldous Huxley</author>

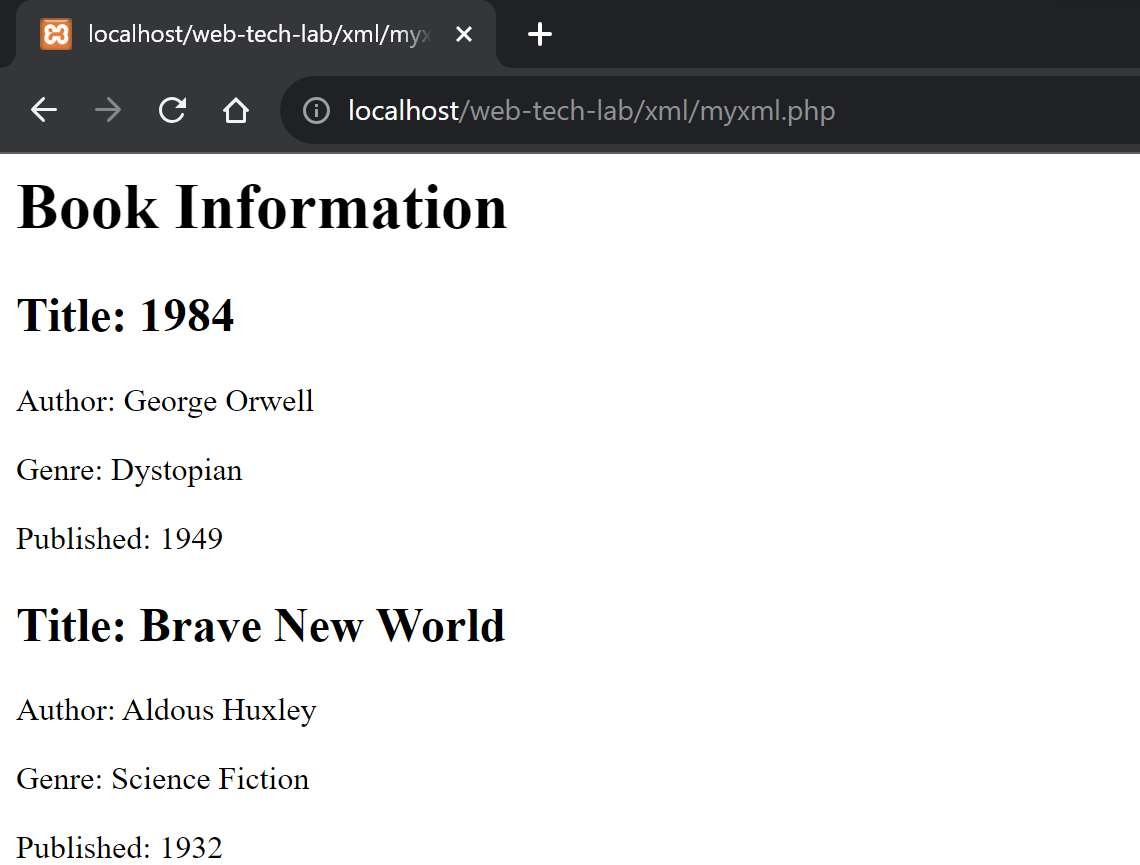
<genre>Science Fiction</genre>

<published>1932</published>

</book>

</library>

**OUTPUT:**



**Practical: 8**

**Aim:** Implement the web applications using JSP.

**Requirement:** Laptop /Computer

## Software Requirement: -

- Windows 10/11 / Browser (Chrome, Brave, Firefox) / Winget

- JSP / XAMPP / VS code / Notepad

**Knowledge Requirement:** HTML, CSS, JSP

**page.jsp**

<html>

<head>

<title>JSP-1</title>

</head>

<body>

<% double num=Math.random(); if (num> 0.95) { %>

<h2>You'll have a luck day!</h2>

<p>(<%= num %>)</p>

<% } else { %>

<h2>Well, life goes on ...</h2>

<p>(<%= num %>)</p>

<% } %>

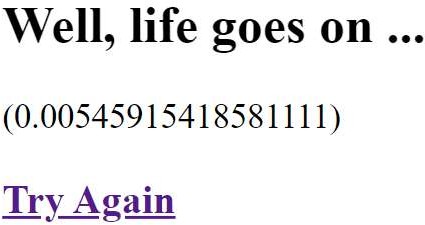
<a href="<%= request.getRequestURI() %>">

<h3>Try Again</h3>

</a>

</body>

</html>





## Index.html

<meta charset="ISO-8859-1" />

<title>This page will send a request to a Java JSP Page.</title>

<a> This page will <b>send a request</b> to a <b>Java JSP Page</b>. </a>

<p style="margin-top: 0; margin-bottom: 0"></p>

<form action="login.jsp">

Name: <input type="text" name="name" />

<br />

Telephone#: <input type="text" name="telephone" />

<input type="submit" value="Submit" />

</form>

A close up of a number

Description automatically generated

## Login.jsp

<title>This is a Java JSP Page - Example</title>

<b><big>This is a Java JSP Page - Example</big></b>

<br>

This is a clasic HTML content into a JSP Page !

<br><br>

From the index.html I have received into JSP Page the <b>name</b> parameter. Name =

<%= request.getParameter("name") %>

<br>

From the index.html I have received into JSP Page the <b>telephone</b> parameter. Telephone# =

<%= request.getParameter("telephone") %>

A close up of a page

Description automatically generated