SHREENIVASA ENGINEERING COLLEGE

(Approved by AICTE and Affiliated to Anna University,Chennai) B.Pallipatti,

Bommidi, Pappireddipatti (Tk), Dharmapuri Dt, Tamilnadu-635 301

DEPARTMENT OF INFORMATION TECHNOLOGY



Bonafide Certificate

This is to certify that this is a Bonafide Record of Practical work ..... done by……of..... Semester of ..... Department in the …….Laboratory During the year…..……

Staff Incharge Head of the Department

Submitted for the University Examination held on …….

Internal Examiner External Examiner

CONTENTS

DATE	EX.NO	LIST OF EXPERIMENTS	PAGE NO	SIGN

EX.NO:1 **CREATION OF INTERACTIVE WEBSITE -**
Date: **DESIGN USING HTML AND AUTHORING TOOLS**

AIM:

The aim of this project is to create an interactive website using HTML and authoring tools.

ALGORITHM:

1. Define the purpose of the website and identify the target audience.

2.Plan the layout of the website, including the pages and navigation.

3.Choose an authoring tool, such as Dreamweaver or WordPress, to create the website.

4.Use HTML to design the pages of the website.

5.Add interactive elements, such as forms and buttons, to engage the user.

6.Optimize the website for search engines and accessibility.

7.Test the website on different devices and browsers.

8.Publish the website to a web hosting service.

PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
   <title>Interactive Webpage</title>
   <style>         button {
background-color:
#4CAF50;            color: white;
padding: 10px 20px;            border:
none;            border-radius: 4px;
cursor: pointer;
       }

       input[type=text] {
padding: 10px;            border:
1px solid #ccc;
border-radius: 4px;
          box-sizing: border-box;
       }
   </style>
</head>

<body>

   <h1>Welcome to My Interactive Webpage!</h1>

   <p>Click the button below to see a message:</p>
<button onclick="alert('Hello World!')">Click Me</button>

   <form>
      <label for="name">Enter Your Name:</label>
      <input type="text" id="name" name="name"><br><br>
      <button onclick="alert('Hello ' +
document.getElementById('name').value)">Say Hello</button>
   </form>
</body>
</html>
```

OUTPUT:


Welcome to My Interactive Webpage!
Click the button below to see a message:
Click Me
Enter Your Name:
Say Hello

RESULT:

Thus the Creation of interactive web sites – Design using HTML and authoring tools have been executed successfully and the output got verified.

EX.NO:2 **FORM VALIDATION USING JAVASCRIPT**

Date:

AIM:

The aim of form validation using JavaScript is to ensure that usersubmitted data is accurate, complete, and properly formatted before it is processed by the server. This helps to prevent errors and data loss, and ensures a better user experience.

ALGORITHMS:

1. Get the form element from the HTML document using JavaScript.

2. Add an event listener to the form element that listens for the submit event.

3.Inside the event listener function, prevent the default form submission behavior using event.preventDefault().

4.Retrieve the values of the form fields using JavaScript and validate them using conditional statements.

5.If any field is invalid, display an error message and prevent the form from submitting.

6.If all fields are valid, submit the form.

PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
        <title>Form Validation Example</title>
        <script>                function validateForm() {
var name = document.forms["myForm"]["name"].value;
var email = document.forms["myForm"]["email"].value;

  if (name == "") {     alert("Name must be filled out");
                        return false;
                }
                if (email == "") {
                        alert("Email must be filled out");
                        return false;
                }
                if (email.indexOf("@") == -1 || email.indexOf(".") == -1) {
                        alert("Please enter a valid email address");
                return false;
                }
        }
        </script>
</head> <body>
        <h1>Form Validation Example</h1>

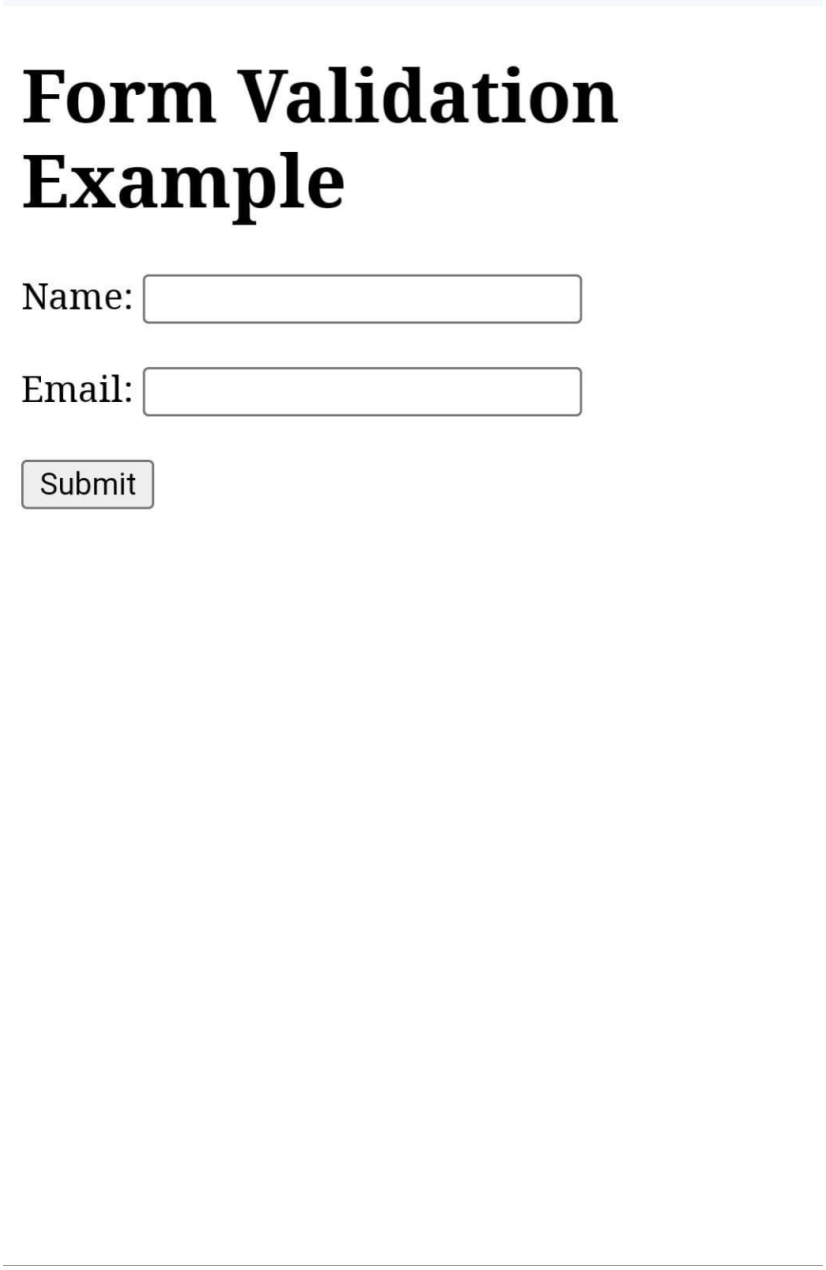
        <form name="myForm" onsubmit="return validateForm()">
 <label for="name">Name:</label>
                <input type="text" id="name" name="name"><br><br>

                <label for="email">Email:</label>
                <input type="text" id="email" name="email"><br><br>

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

</body>
</html>
```

OUTPUT:



RESULT:

Thus the Validation form using Javascript have been executed successfully and the output got verified.

EX.NO:3 CREATION OF SIMPLE PHP SCRIPTS

Date:

AIM:

The aim of creating simple PHP scripts is to enable web developers to build dynamic web pages that can interact with databases and provide customized responses to user input.

ALGORITHMS:

1. Identify the purpose of the PHP script: Determine what the script should accomplish, whether it's processing user input, retrieving data from a database, or generating dynamic content.

2.Set up the PHP environment: Install PHP and configure the web server to support PHP scripts.

3.Create a new PHP file: Use a text editor or integrated development environment (IDE) to create a new file with a .php extension.

4.Start with the PHP opening tag: Begin the PHP script with the opening <?php tag

5.Write PHP code: Write PHP code that performs the desired task, such as processing form data, querying a database, or generating HTML content.

6.Test the PHP script: Save the PHP script to a web server and test it in a web browser to verify that it's functioning correctly.

7.Debug and refine the script: If the script doesn't work as expected, use error messages and debugging tools to identify and fix any issues.

8.Add comments and documentation: Add comments to the code to explain its purpose and make it easier to maintain in the future.

9.Refactor and optimize: Refactor the code to make it more efficient and optimize it for performance.

10.Deploy the PHP script: Upload the PHP script to the web server and integrate it into the website or application as needed.

PROGRAM:

```
<!DOCTYPE html>
<html>
<head>
	<title>Simple PHP Script</title>
</head> <body>

	<h1>Current Date and Time</h1>

	<?php
		echo "The current date and time is: " . date("Y-m-d H:i:s");
	?>

</body> </html>
```

OUTPUT:

Current Date and Time

The current date and time is: 2022-11-11 14:45:30

RESULT:

Thus the Creation of simple PHP scripts have been executed successfully and the output got verified.

EX.NO:4 HANDLING MULTIMEDIA CONTENT IN WEBSITES

Date:

AIM:

The aim of handling multimedia content in web sites is to ensure that users have a seamless and enjoyable experience while accessing multimedia content on a website.

ALGORITHMS:

1.Determine the types of multimedia content that will be used on the website, such as images, videos, audio, and animations.

2.Optimize multimedia files for the web by compressing them without sacrificing quality. This can be done using image compression tools, video compression tools, and audio compression tools.

3.Implement lazy loading for multimedia content to improve the website's loading speed. This involves only loading multimedia content when it is necessary, such as when the user scrolls to a certain point on the page.

4.Use responsive design techniques to ensure that multimedia content is displayed properly on different devices and screen sizes. This includes using responsive image and video tags, and using CSS to adjust the size and position of multimedia elements.

5.Provide an intuitive interface for users to interact with multimedia content, such as allowing them to play and pause videos, adjust the volume, and control the playback speed.

6.Ensure that multimedia content is accessible to users with disabilities, such as providing captions and transcripts for videos and audio files, and using alt tags for images.Monitor website performance and user feedback to continuously optimize multimedia content for the website.

7.Monitor website performance and user feedback to continuously optimize multimedia content for the website.

PROGRAM:

```
    <!DOCTYPE html>
<html>

<head> </head>

<body>
   <img src="https://thumbs.dreamstime.com/b/beautiful-landscape-dry-tree-
branchsunflowers-field-against-colorful-evening-dusky-sky-use-as-natural-
backgroundbackdrop-
48319427.jpg" alt="An example image" /> <br /><br />
   <video controls>
      <source src="video.mp4" type="video/mp4">
<source src="video.webm" type="video/webm">        Your
browser does not support the video tag.
   </video>
   <br /><br />
   <audio controls>
      <source src="audio.mp3" type="audio/mpeg">
<source src="audio.ogg" type="audio/ogg">        Your
browser does not support the audio tag.
   </audio>

</body>

</html>
```

OUTPUT:



RESULT:

Thus the Handling Multimedia content in websites have been executed successfully and the output got verified.

EX.NO:5 i) **WRITE PROGRAMS USING SERVLETS**

Date: **TO INVOKE SERVLETS FROM HTML FORMS**

AIM:

The aim of invoking servlets from HTML forms is to enable the server-side processing of form data submitted by users on a web page.

ALGORITHMS:

1.Create an HTML form that will collect data from the user. The form should have action attribute set to the URL pattern of the servlet that will handle the form submission and a method attribute set to POST or GET.

2.Create a servlet that will handle the form submission. This involves extending the HttpServlet class and overriding the doPost or doGet method, depending on the method used in the form.

3.In the servlet, retrieve the form data submitted by the user using the request object's getParameter method. This method takes the name of the form input element as a parameter and returns its value.

4.Validate the form data to ensure that it meets the required format and values. This involves checking for errors and generating appropriate error messages if necessary.

5.Process the form data to perform the necessary actions, such as updating a database, sending an email, or generating a report.

6.Generate a response to be displayed back to the user. This involves using the response object's PrintWriter to write HTML code that will be displayed in the user's browser.

7.Map the servlet to a URL pattern in the web.xml file. This allows the servlet to be invoked when the form is submitted.

8.Deploy the web application to a server and test the HTML form by submitting data and verifying that the servlet is invoked, the data is processed correctly, and the response is displayed back to the user.

PROGRAM:

HTML:

```
<!DOCTYPE html>
<html>
 <head>
  <title>Form Submission</title>
 </head>
 <body>
  <form action="FormServlet" method="POST">
   <label for="name">Name:</label>
   <input type="text" id="name" name="name"><br>

   <label for="email">Email:</label>
   <input type="email" id="email" name="email"><br>

   <label for="message">Message:</label>
   <textarea id="message" name="message"></textarea><br>

   <input type="submit" value="Submit">
  </form>
</body> </html>
```

SERVLETS:

```
 import java.io.IOException; import
javax.servlet.ServletException; import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest; import
javax.servlet.http.HttpServletResponse;  public class FormServlet
extends HttpServlet {   private static final long serialVersionUID =
1L;

  protected void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {     String name =
request.getParameter("name");
   String email = request.getParameter("email");
String message = request.getParameter("message");

   // Process the form data here (e.g. send an email, save to a database)
```

OUTPUT:

```
    // Generate a response to be displayed back to the user
response.setContentType("text/html");     response.getWriter().println("<html><body>");
    response.getWriter().println("<h2>Thank you for your submission, " + name +
"!</h2>");      response.getWriter().println("<p>We will respond to your message at " +
email + " as
soon as possible.</p>");     response.getWriter().println("</body></html>");
  }
}
```

RESULT:

Thus the given program using Servlets have been executed successfully and the output got verified.

EX.NO:5 II)SESSION TRACKING USING HIDDEN FORM FIELDS
DATE: AND SESSION TRACKING FOR A HIT COUNT

AIM:

To implement session tracking in a Servlet to keep track of the number of times a user has visited a page.

ALGORITHM 1:

Step 1: Create a JSP or HTML page with a form that has a hidden input field. This field should be named "sessionId" and should have a value equal to the session ID.

Step 2: In the Servlet, use the HttpServletRequest object to retrieve the session ID from the hidden input field. If the session ID does not exist in the hidden input field, create a new session ID and store it in the hidden input field.

Step 3: Use the HttpServletResponse object to set a cookie with the session ID.

Step 4: Use the HttpSession object to store any session attributes you need

ALGORITHM 2:

Step 1: Create a Servlet that extends HttpServlet.

Step 2: Override the doGet() method to retrieve the current session, and then retrieve "hitCount" attribute from the session.

Step 3: If the "hitCount" attribute does not exist, set it to 1. Otherwise, increment it by 1.

Step 4: Store the "hitCount" attribute in the session.

Step 5: Send the "hitCount" attribute to the client in the response.

Step 6: Override the doPost() method to call the doGet() method.

PROGRAM:

HTML:

```
<html>
  <body>
    <h2>Enter your name:</h2>
    <form method="post" action="greeting">
      <input type="text" name="name">
      <br><br>
      <input type="hidden" name="sessionid">
      <input type="submit" value="Submit">
    </form>
  </body> </html>
```

SERVLETS:

```
import java.io.*; import
javax.servlet.*;
import javax.servlet.http.*;

public class GreetingServlet extends HttpServlet {
  public void doPost(HttpServletRequest request, HttpServletResponse response)     throws
ServletException, IOException {
response.setContentType("text/html");     PrintWriter out = response.getWriter();

    HttpSession session = request.getSession(true);
    String sessionid = session.getId();
    String name = request.getParameter("name");

    int hitcount = 0;
    if(session.getAttribute("hitcount") != null) {
hitcount = (Integer) session.getAttribute("hitcount");
    }
    hitcount++;
session.setAttribute("hitcount", hitcount);

    out.println("<html>");
out.println("<body>");
    out.println("<h1>Hello, " + name + "!</h1>");     out.println("<p>Your
```

OUTPUT:

```
session ID is " + sessionid + ".</p>");     out.println("<p>You have visited this
page " + hitcount + " times.</p>");     out.println("</body>");
  out.println("</html>");   }
}
```

RESULT:

Thus the given program using Servlets have been executed successfully and the output got verified.

EX.NO:6 **CREATION OF INFORMATION RETREIVEL SYSTEM**
Date: **USING WEB, PHP, AND MYSQL**

AIM:

The aim of this project is to create an information retrieval system using web technologies like PHP and MySQL.

ALGORITHM:

Step 1: Design the database schema for the information retrieval system. Determine the tables, fields, and relationships between the tables.

Step 2: Create a database using MySQL and populate it with sample data.

Step 3: Design the user interface for the search feature. This may involve creating an HTML form where users can enter search terms and submit the form to the server.

Step 4: Write PHP code to handle the user input from the search form. The PHP code should connect to the MySQL database, retrieve the relevant data based on the user input, and display the results to the user.

Step 5: Implement a ranking algorithm to sort the search results by relevance. This may involve calculating a score based on factors like the frequency of the search terms in the database and the proximity of the search terms to each other.

Step 6: Implement pagination to display search results across multiple pages.

Step 7: Add security measures to prevent SQL injection attacks and other security vulnerabilities.

Step 8: Test the system thoroughly to ensure that it works as expected.

Step 9: Deploy the system to a web server and make it accessible to users.

Step 10: Monitor the system for performance and security issues, and make updates as needed to improve the system over time.

PROGRAM:

1.Create a MySQL database and table to store the information we want to retrieve:

```
CREATE DATABASE example_db;

USE example_db;

CREATE TABLE example_table (
   id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(30) NOT NULL,     email VARCHAR(50)
NOT NULL,      phone VARCHAR(20) NOT NULL
);
```

2.Create a PHP script to connect to the database and retrieve the information:

```
<?php
// Connect to database
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "example_db";

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

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

// Retrieve information from database
$sql = "SELECT * FROM example_table";
$result = $conn->query($sql);
// Close database connection
$conn->close();
?>
```

3.Create a web interface using HTML and CSS to display the retrieved information:

3.Create a web interface using HTML and CSS to display the retrieved information:

```
<!DOCTYPE html>
<html>
<head>
   <title>Information Retrieval System</title>
   <style>        table
          border-collapse: collapse;
 width: 100%;
       }        th, td {
 text-align: left;
          padding: 8px;
       }
       tr:nth-child(even) {
          background-color: #f2f2f2;
       }
   </style>
</head>
<body>
   <h1>Information Retrieval System</h1>
   <table>
       <tr>
          <th>ID</th>
          <th>Name</th>
          <th>Email</th>
          <th>Phone</th>
       </tr>
       <?php
       // Output retrieved information in table rows
if ($result->num_rows > 0) {
 while($row
= $result->fetch_assoc()) {
 echo "<tr>";
echo "<td>" . $row["id"] . "</td>";
 echo "<td>" . $row["name"] . "</td>";
 echo "<td>" . $row["email"] . "</td>";
echo "<td>" . $row["phone"] . "</td>";
echo "</tr>";
```

OUTPUT:

```
ID   | Name     | Email                  | Phone
-----------------------------------------------------
1    | John Doe | johndoe@example.com       | 555-1234
2    | Jane Doe | janedoe@example.com       | 555-5678 3
     | Bob Smith| bob.smith@example.com     | 555-9876
```

RESULT:

Thus the given program Creation of retreivel System using web, PHP, and MySQL have been executed successfully and the output got verified

EX.NO:7 **CREATION OF INFORMATION RETREIVEL SYSTEM**
Date: **USING WEB, PHP, AND MYSQL**

AIM:

The aim of creating a Personal Information System (PIS) is to provide individuals with a centralized platform to manage their personal information securely and conveniently.

ALGORITHM:

1.Define the data types to be included in the PIS such as personal details (name, date of birth, contact details, etc.), education details, work experience, financial information, medical history, etc.

2.Design the user interface for the PIS. The interface should be intuitive and userfriendly. It should provide easy navigation and access to different sections of the PIS.

3.Develop a secure login system with multi-factor authentication to protect the user's data from unauthorized access.

4.Create a database to store user data. The database should be secure, scalable, and easily accessible. Use encryption and hashing techniques to secure the data.

5.Implement data validation and verification methods to ensure the accuracy and integrity of the data entered by the user.

6.Create an analytics engine to generate insights and reports based on the user's data. This engine can provide valuable information on the user's spending habits, health status, career progress, etc.

7.Enable data sharing with trusted third-party services if required. The user should have full control over what data is shared and with whom.

8.Provide data backup and recovery mechanisms to ensure that the user's data is not lost in case of hardware failure or other unforeseen events.
9.Test the PIS thoroughly to ensure that it meets the requirements and specifications.

10.Roll out the PIS and provide ongoing support and maintenance to ensure its smooth operation.

PROGRAM:

1.Database setup:
Start by creating a MySQL database and a table named "users". The "users" table will store the user's personal information.

```
CREATE DATABASE personal_info;

USE personal_info;

CREATE TABLE users (
  id INT(11) NOT NULL AUTO_INCREMENT PRIMARY KEY,
name VARCHAR(255) NOT NULL,   address VARCHAR(255)
NOT NULL,   phone VARCHAR(255) NOT NULL,    email
VARCHAR(255) NOT NULL
);
```

2.**HTML form**:
Create an HTML form that allows users to input their personal information.

```
<form action="add_user.php" method="post">
  <label>Name:</label>
  <input type="text" name="name" required>
  <label>Address:</label>
  <input type="text" name="address" required>
  <label>Phone:</label>
  <input type="text" name="phone" required>
  <label>Email:</label>
  <input type="email" name="email" required>
  <button type="submit">Submit</button>
</form>
```

3.**PHP script to add user**:

3.**PHP script to add user**:

Create a PHP script named "add_user.php" that adds the user's information to the "users" table in the database.

```
<?php
// connect to the database
$servername = "localhost";
$username = "username";
$password = "password"; $dbname
= "personal_info";

$conn = mysqli_connect($servername, $username, $password, $dbname);

// check connection if (!$conn) {   die("Connection failed:
" . mysqli_connect_error());
}

// insert user data into the database
$name = $_POST['name']; $address
= $_POST['address']; $phone =
$_POST['phone'];
$email = $_POST['email'];

$sql = "INSERT INTO users (name, address, phone, email) VALUES ('$name',
'$address', '$phone', '$email')";

if (mysqli_query($conn, $sql)) {
echo "User added successfully";
} else {
  echo "Error: " . $sql . "<br>" . mysqli_error($conn);  }

mysqli_close($conn);
?>
```

4.**PHP script to display user information**:

Create a PHP script named "view_users.php" that displays all users' information from the "users" table in the database.

```
$servername = "localhost";
$username = "username";
$password = "password";
$dbname = "personal_info";

$conn = mysqli_connect($servername, $username, $password, $dbname);

// check connection if (!$conn) {   die("Connection failed:
" . mysqli_connect_error());
}

// get all users from the database
$sql = "SELECT * FROM users";
$result = mysqli_query($conn, $sql);

// display user data  if
(mysqli_num_rows($result) > 0) {
while($row = mysqli_fetch_assoc($result)) {
echo "Name: " . $row["name"]. "<br>";     echo
"Address: " . $row["address"]. "<br>";     echo
"Phone: " . $row["phone"]. "<br>";
    echo "Email: " . $row["email"]. "<br><br>";
  }
} else {
  echo "No users found.";
}
mysqli_close($conn);
?>
```

RESULT:

Thus the given program Creation of Personal Information System have been executed successfully and the output got verified.