

INDEX

Exp.No	Date	Name of the Experiment	Page No.	Staff Signature
1.		HTML Page with Image, Input Controls, and Anchor Tag		
2.		Designing a Web Page with Table, Lists, and Table Hover Effect		
3.		HTML Program to use Image Map		
4.		Webpage using Cascading Style Sheet		
5.		Validating Web Form controls using DHTML		
6.		Develop a Webpage Using Scripting Languages with the Help of CSS		
7.		Generate the Fibonacci series using PHP user-defined function		
8.		Form Handling in PHP- Create a recruitment website where a job seeker can upload his/her details		
9.		Create an Employee database with two fields Employer's Name, Employee's Name with MySql and insert two records into those fields using PHP CODE.		
10.		Apply any two PHP sort functions each on an indexed array and an associative array		

HTML Page with Image, Input Controls, and Anchor Tag

AIM:

To create a simple web page using HTML that demonstrates the use of image, checkbox inputs, radio button inputs, and anchor tags.

ALGORITHM:

1. Start the HTML document using the `<!DOCTYPE html>` declaration.
2. Create the basic structure of the web page using the `<html>`, `<head>`, and `<body>` elements.
3. Inside the `<body>` section, insert an image using the `<img>` tag and set the source attribute to "logo.svg".
4. Add at least two checkbox inputs. The labels or text for the checkboxes can be of the user's choice.
5. Add at least two radio button inputs. The labels or options for the radio buttons can be of the user's choice.
6. Create an anchor element using the `<a>` tag that links to: <https://drmgrdu.codetantra.com/>. The text displayed for the link can be of the user's choice. Add the `target="_blank"` attribute.
7. Close all the opened HTML tags properly.

PROGRAM:

```
<!DOCTYPE html>
<html>

<head>
  <title>Simple Web Page</title>
</head>

<body>

  <!-- Image -->
  

  <br><br>

  <!-- Checkboxes -->
  <label>
    <input type="checkbox" name="option1"> Option 1
  </label>
  <br>
  <label>
    <input type="checkbox" name="option2"> Option 2
  </label>

  <br><br>

  <!-- Radio Buttons -->
  <label>
    <input type="radio" name="gender" value="male"> Male
  </label>
  <label>
```

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

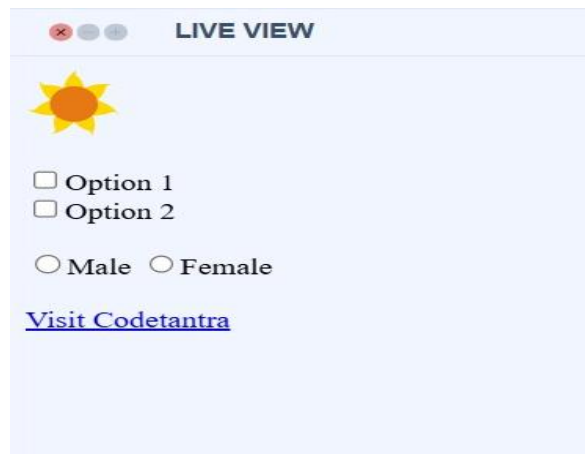
<br><br>

<!-- Anchor -->
<a href="https://drmgrdu.codetantra.com/" target="_blank">
  Visit Codetantra
</a>

</body>

</html>
```

Output:



Result :

Thus, the HTML program to create a webpage that includes an image, checkboxes, radio buttons, and a hyperlink is executed successfully, and the output is verified.

Designing a Web Page with Table, Lists, and Table Hover Effect

Aim:

To create a web page using HTML that demonstrates the use of tables, different types of lists, and a CSS hover effect applied to table rows.

Algorithm:

1. Begin the document with the `<!DOCTYPE html>` declaration.
2. Create the basic structure using `<html>`, `<head>`, and `<body>` tags.
3. Inside the `<head>` section:
 - o Insert a `<style>` element to define internal CSS.
 - o Inside the `<style>` element, define a CSS rule using the `tr:hover` selector to change the background color of table rows to `#add8e6` when the mouse pointer hovers over them.
4. Inside the `<body>` section:
 - o Create a table using the `<table>` element with at least one row and one column.
 - o Add an ordered list using the `<ol>` element with at least one list item `<li>`.
 - o Add an unordered list using the `<ul>` element with at least one list item `<li>`.
 - o Add a description list using the `<dl>` element with at least one term `<dt>` and one description `<dd>`.

Program:

```
<!DOCTYPE html>
<html>

<head>
  <title>Table and Lists</title>

  <!-- Write your code here -->
  <style>
    tr:hover {
      background-color: #add8e6;
    }
  </style>
</head>

<body>

  <!-- Write your code here -->

  <h2>Sample Table</h2>
  <table border="1">
    <tr>
      <th>Product</th>
```

```
    <th>Price</th>
</tr>
<tr>
    <td>Laptop</td>
    <td>50000</td>
</tr>
</table>
```

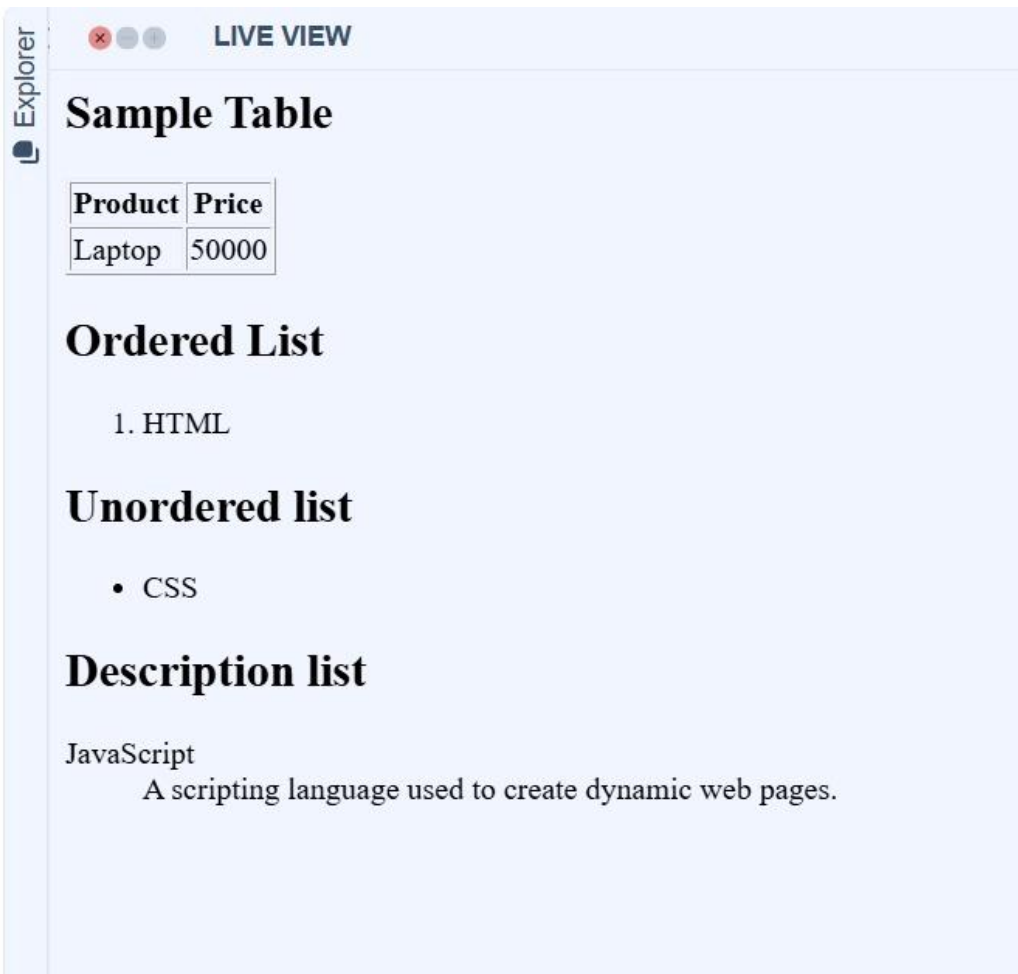
```
<h2>Ordered List</h2>
<ol>
    <li>HTML</li>
</ol>
```

```
<h2>Unordered List</h2>
<ul>
    <li>CSS</li>
</ul>
```

```
<h2>Description List</h2>
<dl>
    <dt>JavaScript</dt>
    <dd>A scripting language used to create dynamic web pages.</dd>
</dl>
```

```
</body>
</html>
```

Output:



Result:

Thus, the HTML program to create a webpage that includes a table, types of lists (unordered, ordered, and description), and uses the hover effect on an element is executed successfully, and the output is

EX.NO: 03

DATE :

HTML Program to use Image Map

Aim:

To create a web page using HTML that demonstrates embedding an image map using the provided image file logo.svg, defining hotspot regions, and linking the hotspots to relevant information when clicked.

Algorithm:

1. Begin the document with the <!DOCTYPE html> declaration and create the basic structure using <html>, <head>, and <body> tags.
2. Inside the <body> section:
 - Insert an element with the source set to "logo.svg".
 - Use the usemap attribute in the element to associate the image with a map named "logomap".
3. Create a <map> element and assign it the name "logomap" using the name attribute.
4. Inside the <map> element, define at least two hotspot regions using <area> elements.
5. For each <area> element:
 - Specify a shape attribute (such as rect, circle, or poly) of your choice.
 - Specify appropriate coords values to define the clickable region.
 - Add the href attribute and set it to <https://drmgrdu.codetantra.com/>.
 - Add the target attribute and set it to "_blank".

Program:

```
<!DOCTYPE html>
<html>

<head>
  <title>HTML using Image Map</title>
  <link rel="stylesheet" href="style.css">
</head>

<body>

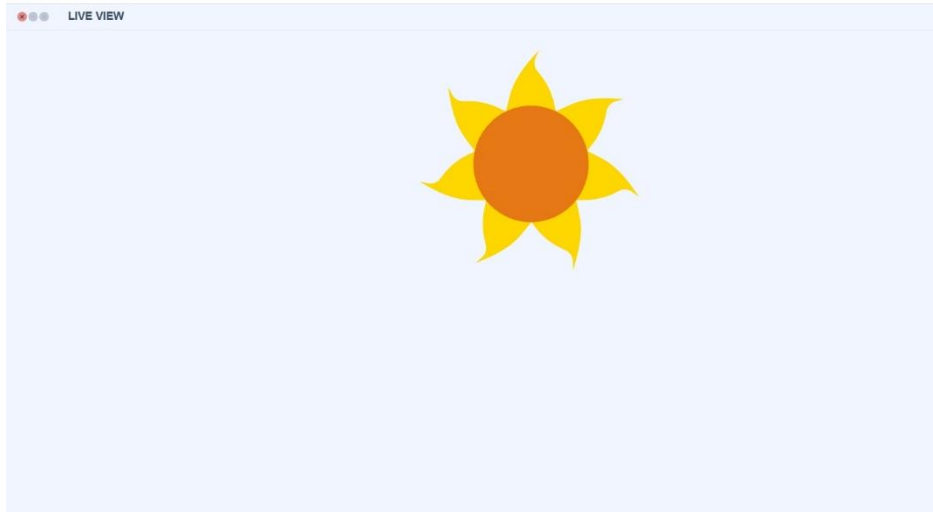
  <map name="logomap">
    <area shape="rect" coords="0,0,500,250"
      href="https://drmgrdu.codetantra.com/" target="_blank">

    <area shape="rect" coords="500,0,1000,500"
      href="https://drmgrdu.codetantra.com/" target="_blank">
  </map>

</body>

</html>
```

OUTPUT:



Result:

Thus, the HTML program to create a webpage using an image map with multiple hotspot regions that navigate to the specified webpage when clicked is executed successfully, and the output is verified.

Webpage using Cascading Style Sheet

Aim:

To create a web page that demonstrates the use of all three types of Cascading Style Sheets: External, Internal (Embedded), and Inline CSS.

Algorithm:

1. Create an HTML document with the proper structure using `<!DOCTYPE html>`, `<html>`, `<head>`, and `<body>` elements.
2. In the head section, link the external stylesheet `MyStyle.css` using the `<link>` tag.
3. Add an embedded stylesheet inside the `<style>` tag in the `<head>` section.
4. Inside the embedded stylesheet, define the following style:
 - Set the left margin of the `h1` element to `40px`.
5. In the external stylesheet (`MyStyle.css`), define the following styles:
 - Set the background color of the body to `#ADD8E6`.
 - Set the text color of the `h1` element to `#000080`.
6. In the `<body>` section, include an `<h1>` heading element with any text.
7. Add a paragraph `<p>` element below the heading.
8. Apply inline CSS to the paragraph element by:
 - Setting the text color to `#0000FF`.
 - Setting the top margin to `30px`.

PROGRAM:**HTML**

```
<!DOCTYPE html>
<html>
<head>
  <title>CSS Types Demonstration</title>

  <!-- Link external CSS -->
  <link rel="stylesheet" href="MyStyle.css">

  <!-- Embedded CSS -->
  <style>
    h1 {
      margin-left: 40px;
    }
  </style>

</head>
<body>

  <h1>This is Heading</h1>

  <!-- Paragraph with inline CSS -->
  <p style="color:#0000FF; margin-top:30px;">
```

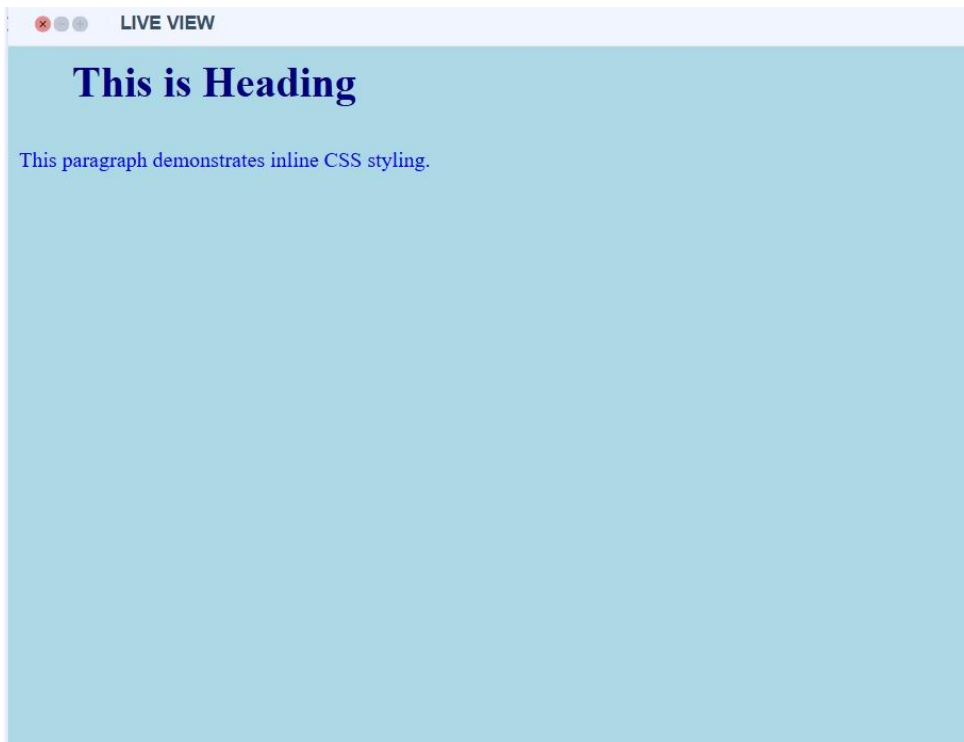
This paragraph demonstrates inline CSS styling.

</body>
</html>

Style.css

```
/* Add your external CSS here */  
  
body {  
    background-color: #ADD8E6;  
}  
  
h1 {  
    color: #000080;  
}
```

OUTPUT:



Result:

The web page was created successfully using External, Internal (Embedded), and Inline CSS, and all the required styles were applied correctly.

Validating Web Form controls using DHTML**Aim:**

To write a client-side script using JavaScript (DHTML) to validate a web form field and display validation messages beside the input field.

Algorithm:

1. Create an HTML document with the proper structure using `<!DOCTYPE html>`, `<html>`, `<head>`, and `<body>` elements.
2. Inside the `<body>` section, create a `<form>` element with the name `contact_form`.
3. Inside the form, add:
 - A text input field with the name `contact_name`.
 - A placeholder element such as a `<span>` with an id `msg` below the input field to display validation messages.
 - A submit button.
4. Inside the `<head>` section, add a `<script>` block.
5. Inside the script block, define a function named `validate_form()`.
6. Inside the function:
 - Declare a variable `valid` to store the validation status.
 - Declare a variable `name` and assign it the value of the `contact_name` input field of the form.
 - Declare a variable `message` and assign it the placeholder element using: `document.getElementById("msg")`
7. Check whether the name variable is empty.
8. If the name is empty:
 - Display the message "Please fill in the name" in the message element.
 - Set the variable `valid` to false.
9. If the name is not empty:
 - Display the message "Successful" in the message element.
 - Set the variable `valid` to true.
10. Return the value of the `valid` variable from the function `validate_form()`.

Program:

```
<!DOCTYPE html>
<html>
<head>
  <title>A Simple Form with JavaScript Validation</title>

  <script type="text/javascript">
    function validate_form() {

      // Write your code here...
      var valid;

      // Get the name value from the form
      var name = document.contact_form.contact_name.value;

      // Get the message placeholder
      var message = document.getElementById("msg");

      // Check if name is empty
      if (name == "") {

        // Display error message
        message.innerHTML = "Please fill in the name";

        // Set validation to false
        valid = false;

      } else {

        // Display success message
        message.innerHTML = "Successful";

        // Set validation to true
        valid = true;

      }

      // Return validation status
      return valid;
    }
  </script>

</head>
<body>

  <form name="contact_form" method="post" onsubmit="return validate_form();">

    <h1>Please Enter Your Name</h1>
```

```
<p>
  Your Name:
  <input type="text" name="contact_name">
</p>
```

```
<!-- Placeholder for validation message -->
<span id="msg"></span>
```

```
<p>
  <input type="submit" name="send" value="Send Details">
</p>
```

```
</form>
```

```
</body>
</html>
```

Output:



A screenshot of a web browser window titled "LIVE VIEW". The page has a light blue background. At the top, the text "Please Enter Your Name" is displayed in a large, bold, black serif font. Below this, the text "Your Name:" is followed by a white text input field with a thin black border. Underneath the input field is a button with a light gray background and a thin black border, containing the text "Send Details" in a small, black, sans-serif font.

Result:

The web form was successfully validated using JavaScript (DHTML), displaying appropriate validation messages based on the user input.

Develop a Webpage Using Scripting Languages with the Help of CSS

Aim:

To implement an Interactive Product Price Calculator using JavaScript that dynamically calculates and displays the total bill based on product selection, quantity, and additional service options.

Algorithm:

1. In the external stylesheet styles.css, apply the following styles:
 - Set the background color of the body to #f4f6f9.
 - For the .container class, set:
 - background-color to #ffffff
 - padding to 20px
 - Style the button element by setting:
 - background-color to #1e3a8a
 - color to #ffffff
2. Create a JavaScript function named calculateTotal().
3. Inside the function, read the selected product price from the dropdown element with id product using its value attribute.
4. Read the quantity value from the input field with id quantity using its value attribute.
5. Convert both product price and quantity to numeric values.
6. Calculate the initial total by multiplying:
 - product price $\times$ quantity
7. Check whether the checkbox with id warranty is selected using the checked attribute.
 - If selected, add 2000 to the total amount.
8. Check whether the checkbox with id delivery is selected using the checked attribute.
 - If selected, add 500 to the total amount.
9. Display the final total amount inside the element with id total using the innerText property.
10. When the "Calculate Total" button is clicked, the calculateTotal() function should execute, compute the total price, and display the result on the page

Program :

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="author" content="Pranay Patil">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Product Price Calculator</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>Electronics Shop - Price Calculator</h1>

<div class="container">
```

```

<!-- Product Selection -->
<label>Select Product:</label>
<select id="product">
  <option value="0">--Select--</option>
  <option value="50000">Laptop - ₹50,000</option>
  <option value="20000">Smartphone - ₹20,000</option>
  <option value="1500">Headphones - ₹1,500</option>
</select>

<!-- Quantity -->
<label>Quantity:</label>
<input type="number" id="quantity" value="1" min="1">

<!-- Additional Options -->
<label>Add-ons:</label><br>
<input type="checkbox" id="warranty"> Extended Warranty (₹2000)<br>
<input type="checkbox" id="delivery"> Home Delivery (₹500)<br>

<!-- Button -->
<button onclick="calculateTotal()">Calculate Total</button>

<!-- Output -->
<h2>Total Price: ₹<span id="total">0</span></h2>

</div>

<script src="script.js"></script>

</body>
</html>

```

Styles.css-

```

/* Add your external CSS here */

/* Set background color for the page */
body {
  background-color: #f4f6f9;
}

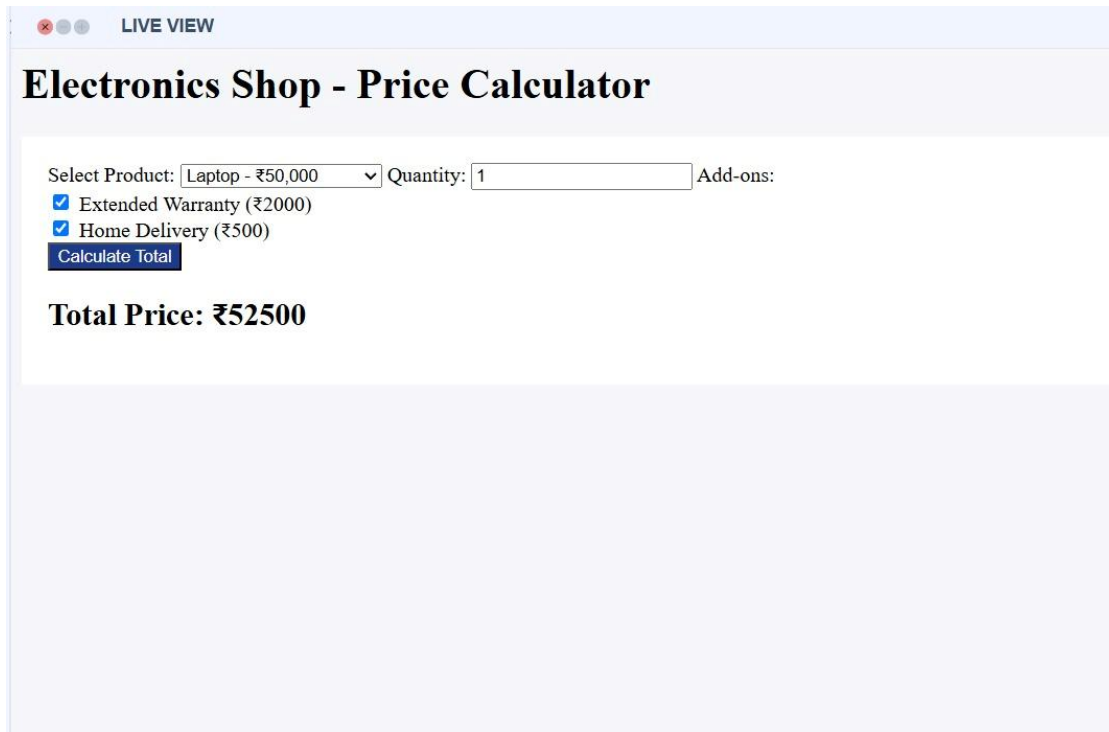
/* Container styling */
.container {
  background-color: #ffffff;
  padding: 20px;
}

/* Button styling */
button {
  background-color: #1e3a8a;
  color: #ffffff;
}

```

}

Output:



The screenshot shows a web browser window with a title bar that says "LIVE VIEW". The main heading of the page is "Electronics Shop - Price Calculator". Below the heading, there is a form with the following elements:

- "Select Product:" followed by a dropdown menu showing "Laptop - ₹50,000".
- "Quantity:" followed by a text input field containing the number "1".
- "Add-ons:" followed by two checked checkboxes:
 - Extended Warranty (₹2000)
 - Home Delivery (₹500)
- A blue button labeled "Calculate Total".
- Below the form, the text "Total Price: ₹52500" is displayed.

Result:

The product price calculator was successfully developed using HTML, CSS, and JavaScript, and it correctly calculates and displays the total price based on the selected options.

Generate the Fibonacci series using PHP user-defined function**Aim:**

To generate the Fibonacci series using a user-defined function in PHP.

Algorithm:

1. Create a PHP file to define a user-defined function for generating the Fibonacci series.
2. In the function, initialize the first two terms of the Fibonacci sequence as 0 and 1.
3. Use a loop to calculate the next terms in the series.
4. Display the Fibonacci series for a given number of terms (input by the user).
5. Save and execute the PHP file in XAMPP to verify the result.

Program:

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Fibonacci Series</title>

</head>

<body>

<h1>Generate Fibonacci Series using PHP</h1>

<?php

// User-defined function to generate Fibonacci series function generateFibonacci($n) {
$first = 0;

$second = 1;
```

```

// Display the first two numbers of the Fibonacci series echo $first . " " . $second . " ";

// Loop to calculate the next terms in the series for ($i = 3; $i <= $n; $i++) {
$next = $first + $second; echo $next . " ";

// Update the values for the next iteration

$first = $second;

$second = $next;

}

}

// Input: Number of terms to display

$terms = 10; // Change this value to generate a different number of terms


// Call the function and display the Fibonacci series echo "<h3>Fibonacci Series (first $terms terms):</h3>";
generateFibonacci($terms);
?>

</body>

</html>

```

OUTPUT:

```
Generate Fibonacci Series using PHP
```

```
Fibonacci Series (first 10 terms):
```

```
0 1 1 2 3 5 8 13 21 34
```

Result:

Thus, the PHP program to generate the Fibonacci series using a user-defined function is executed successfully, and the output is verified.

Form Handling in PHP- Create a recruitment website where a job seeker can upload his/her details

Aim:

To create a recruitment website using PHP where a job seeker can submit their personal details and upload their resume. The form will handle user input and the file upload process.

Algorithm:

1. Create an HTML form for job seekers to submit their personal details (name, email, phone number, etc.) and upload their resume.
2. The form data will be sent to a PHP script for processing.
3. In PHP, validate the form data and handle the file upload securely.
4. Store the job seeker's information in variables and upload the resume to a designated folder.
5. Display a success message once the form has been successfully submitted.

Program :

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Job Seeker Recruitment Form</title>

  <style>
    body {
      font-family: Arial, sans-serif;
      background-color: #f7f7f7;
      margin: 0;
      padding: 20px;
    }

    h1 {
      text-align: center;
      color: #333;
    }

    .form-container {
      width: 50%;
      margin: auto;
      background-color: #fff;
      padding: 30px;
      box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
    }
```

```

    border-radius: 10px;
}

label {
    font-weight: bold;
    margin-bottom: 10px;
    display: block;
}

input[type="text"], input[type="email"], input[type="tel"], input[type="file"] {
    width: 100%;
    padding: 10px;
    margin: 10px 0;
    border: 1px solid #ccc;
    border-radius: 5px;
}

input[type="submit"] {
    background-color: #4CAF50;
    color: white;
    padding: 12px 20px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    width: 100%;
}

input[type="submit"]:hover {
    background-color: #45a049;
}

.error {
    color: red;
    font-size: 12px;
}

.success {
    color: green;
    font-size: 14px;
    text-align: center;
}
</style>
</head>
<body>

<h1>Job Seeker Recruitment Form</h1>

<div class="form-container">
    <form action="submit.php" method="POST" enctype="multipart/form-data">

```

```

<label for="name">Full Name:</label>
<input type="text" id="name" name="name" required>

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

<label for="phone">Phone Number:</label>
<input type="tel" id="phone" name="phone" required>

<label for="resume">Upload Resume (PDF, DOC, DOCX):</label>
<input type="file" id="resume" name="resume" accept=".pdf,.doc,.docx" required>

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

</body>
</html>

```

submit.php

```

<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    // Get form data
    $name = $_POST['name'];
    $email = $_POST['email'];
    $phone = $_POST['phone'];

    // Validate form fields
    if (empty($name) || empty($email) || empty($phone)) {
        echo "<p class='error'>All fields are required.</p>";
        exit;
    }

    // File upload handling
    $resume = $_FILES['resume'];
    $resume_name = $resume['name'];
    $resume_tmp = $resume['tmp_name'];
    $resume_size = $resume['size'];
    $resume_error = $resume['error'];

    // File upload validation
    $allowed_types = ['pdf', 'doc', 'docx'];
    $resume_ext = pathinfo($resume_name, PATHINFO_EXTENSION);

    if ($resume_error !== 0) {
        echo "<p class='error'>Error in uploading the file.</p>";
        exit;
    }
}

```

```

if (!in_array($resume_ext, $allowed_types)) {
    echo "<p class='error'>Only PDF, DOC, and DOCX files are allowed.</p>";
    exit;
}
if ($resume_size > 5000000) { // Limit to 5MB
    echo "<p class='error'>File size should not exceed 5MB.</p>";
    exit;
}
// Define target directory
$upload_dir = 'uploads/';
if (!file_exists($upload_dir)) {
    mkdir($upload_dir, 0777, true);
}
// Generate unique filename to avoid conflicts
$resume_new_name = uniqid("", true) . '.' . $resume_ext;
$resume_path = $upload_dir . $resume_new_name;
// Move the uploaded file to the target directory
if (move_uploaded_file($resume_tmp, $resume_path)) {
    echo "<p class='success'>Application submitted successfully!</p>";
    echo "<p class='success'>Name: $name</p>";
    echo "<p class='success'>Email: $email</p>";
    echo "<p class='success'>Phone: $phone</p>";
    echo "<p class='success'>Resume uploaded successfully: <a href='$resume_path' target='_blank'>View
Resume</a></p>";
} else {
    echo "<p class='error'>Failed to upload resume.</p>";
}
}
?>

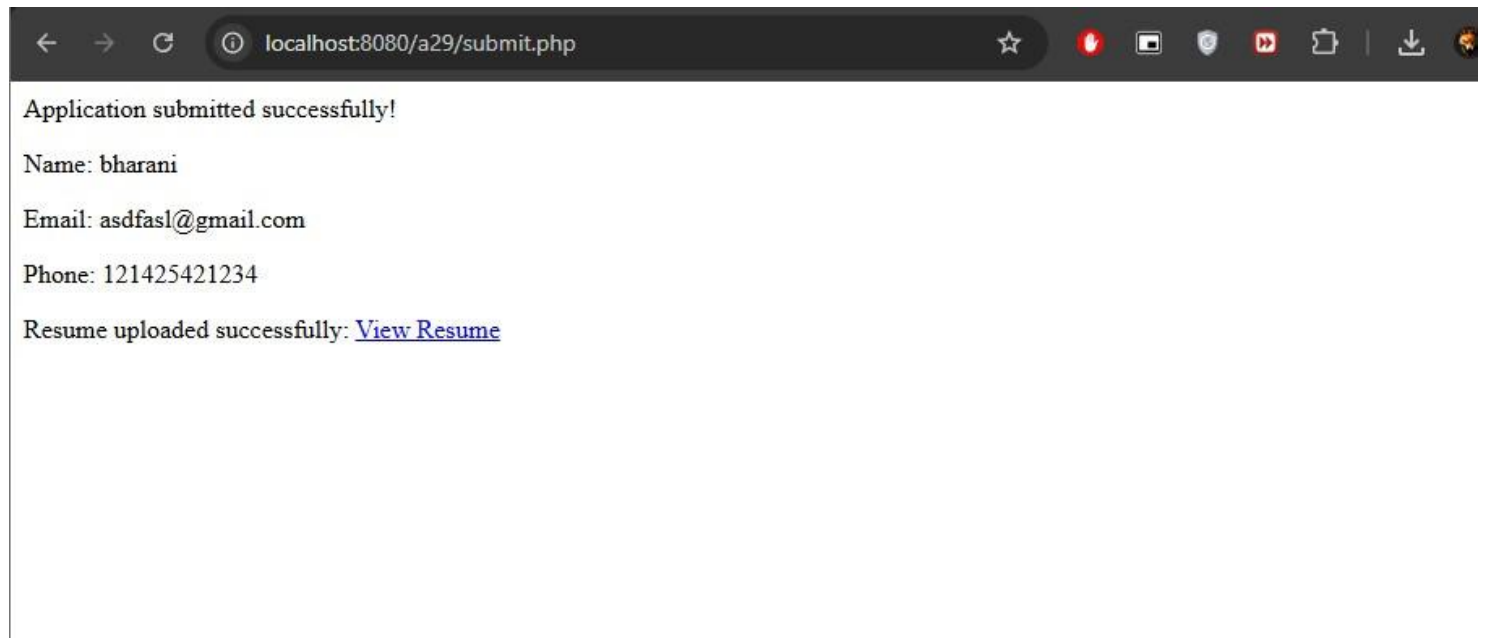
```

Output:

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/a29/'. The main content area features a 'Job Seeker Recruitment Form' with the following fields and values:

- Full Name:** bharani
- Email:** asdfast@gmail.com
- Phone Number:** 121425421234
- Upload Resume (PDF, DOC, DOCX):** A file named 'DocScanner4Jul202411-06.pdf' is selected.

A green 'Submit Application' button is located at the bottom of the form.



Result:

When the job seeker submits the form:

- If all inputs are valid and the file upload succeeds, a success message will be displayed with the job seeker's details and a link to view the uploaded resume.
- If there are any issues (empty fields, incorrect file type, or file size), appropriate error messages will be shown.

EX.NO: 09

DATE :

Create an Employee database with two fields Employer's Name, Employee's Name with MySql and insert two records into those fields using PHP CODE.

Aim:

To create a database in MySQL to store employer and employee information and use PHP code to insert two records into the database.

Algorithm:

1. Start
2. Assign server name, user name, password, dbname , conn variables with therequired values.
3. Check if the connection was successful. If not, print connection failed, else, proceedto step (4).
4. Insert a record and assign that SQL query to sql object.
5. Check the execution of the query with the conn object. If successful, print "recordinserted successfully" else print "error".
6. Close the connection.
7. Stop.
- 8.

Program :

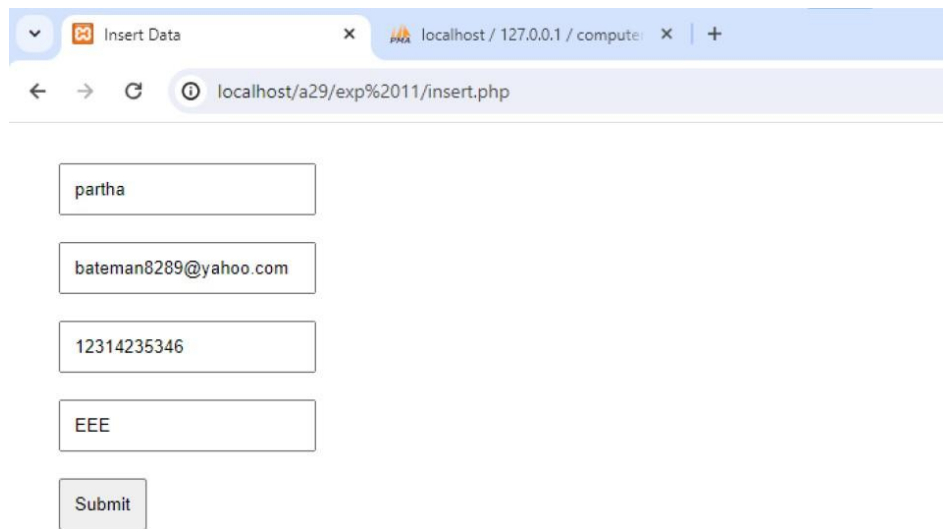
insert.php

```
<html>
<style> form,input
{
padding:10px;margin:10px;
}
</style>
<body>
<title>Insert Data</title>
<form action="Main.php" method="POST">
<input type="text" name="Name" placeholder="Name"><br>
<input type="email" name="Email" placeholder="Email Id"><br>
<input type="mobile" name="Mobile" placeholder="Mobile-number"><br>
<input type="text" name="Department" placeholder="Department"><br>
<input type="submit" name="submit">
</form>
</body>
</html>
```

Main.php

```
<?php
$con = mysqli_connect('localhost','root',''); if(!$con)
{
echo "Not Connected to server";
}
if (!mysqli_select_db($con,'computer'))
{
echo "Database is not selected";
}
$name = $_POST['Name'];
$email = $_POST['Email'];
$mobile = $_POST['Mobile'];
$department = $_POST['Department'];
$sql="INSERT INTO student(Name,Email,Mobile,Department) VALUES
('$name','$email','$mobile','$department')";
if (!mysqli_query($con,$sql))
{
echo "Failed";
}
else
{
echo "Passed";
}
header("refresh:2; url=insert.php");
?>
```

Output:



The screenshot shows a web browser window with two tabs. The active tab is titled 'Insert Data' and shows a form with four text input fields and a submit button. The first field contains 'partha', the second contains 'bateman8289@yahoo.com', the third contains '12314235346', and the fourth contains 'EEE'. The submit button is labeled 'Submit'. The browser's address bar shows the URL 'localhost/a29/exp%2011/insert.php'.

Server: 127.0.0.1 » Database: computer » Table: details

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 9 (10 total, Query took 0.0006 seconds.)

```
SELECT * FROM `details`
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Extra options

Name	Email	Mobile	Department
partha	sfasdf@gmail.com	2147483647	CSE department
batman	sfasdf@gmail.com	2147483647	CSE department
		0	
		0	
Annamalai	siva.annamalai@gmail.com	1234567890	CSE
Annamalai	siva.annamalai@gmail.com	1234567890	CSE
		0	
		0	
		0	
partha	kritisanon9299@yahoo.com	2147483647	EEE

☐ Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

Print | Copy to clipboard | Export | Display chart | Create view

Result:

Hence, the 'PHP/MySQL' program to insert a record into a database table is executed and the output verified.

EX.NO: 10

DATE :

Apply any two PHP sort functions each on an indexed array and an associative array

Aim:

To apply any two PHP sort functions, each on an indexed array and an associative array.

Algorithm:

1. Create a PHP file to define an indexed array and an associative array.
2. Apply sorting functions such as `sort()`, `rsort()` for indexed arrays, and `asort()`, `ksort()` for associative arrays.
3. Display the original and sorted arrays to observe the changes.
4. Save and execute the PHP file in XAMPP to verify the result.

PROGRAM:

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

  <h1> Apply PHP Sort Functions</h1>

  <?php
  // Indexed array
  $indexedArray = array(10, 3, 45, 7, 29, 1);

  // Associative array
  $associativeArray = array("John" => 25, "Jane" => 30, "Peter" => 22, "Tom" => 35);

  // Display the original indexed array
  echo "<h3>Original Indexed Array:</h3>";
  print_r($indexedArray);

  // Applying sort() on indexed array (ascending order)
  sort($indexedArray);
  echo "<h3>Indexed Array after sort() (ascending):</h3>";
  print_r($indexedArray);

  echo "<br>";

  // Display the original associative array
  echo "<h3>Original Associative Array:</h3>";
```

```

print_r($associativeArray);

// Applying asort() on associative array (sort by values)
asort($associativeArray);
echo "<h3>Associative Array after asort() (sorted by values):</h3>";
print_r($associativeArray);

echo "<br>";

// Applying ksort() on associative array (sort by keys)
ksort($associativeArray);
echo "<h3>Associative Array after ksort() (sorted by keys):</h3>";
print_r($associativeArray);
?>
</body>
</html>

```

OUTPUT:

```

Original Indexed Array:
Array ( [0] => 10 [1] => 3 [2] => 45 [3] => 7 [4] => 29 [5] => 1 )

Indexed Array after sort():
Array ( [0] => 1 [1] => 3 [2] => 7 [3] => 10 [4] => 29 [5] => 45 )

Original Associative Array:
Array ( [John] => 25 [Jane] => 30 [Peter] => 22 [Tom] => 35 )

Associative Array after asort():
Array ( [Peter] => 22 [John] => 25 [Jane] => 30 [Tom] => 35 )

Associative Array after ksort():
Array ( [Jane] => 30 [John] => 25 [Peter] => 22 [Tom] => 35 )

```

Result:

Thus, the PHP program to apply two sort functions on an indexed array and an associative array is executed successfully, and the output is verified.