**Experiment Documentation: PHP Form Example**

**Experiment Title: PHP Form Handling**

**1. Problem Statement**

**To demonstrate form handling in PHP by creating a simple form that accepts user input (name and email) and processes it upon form submission.**

**2. Steps and Explanation**

1. **HTML Document Declaration:**
   * **The document starts with <!DOCTYPE html> declaring the HTML5 document type.**
   * **The <html> tag includes the lang="en" attribute, specifying the language as English.**
2. **Head Section:**
   * **The <meta charset="UTF-8"> tag ensures that the document handles all characters correctly.**
   * **The <meta name="viewport" content="width=device-width, initial-scale=1.0"> ensures the page is responsive, scaling properly on different devices.**
   * **The CSS <style> section provides styling for the form:**
     + **Form Container: A .form-container class is used to style the form box with a width of 300px, center alignment, padding, and a light background.**
     + **Input Fields: The input[type="text"] and input[type="email"] elements are styled for width, padding, and margin, creating a neat, user-friendly form interface.**
     + **Submit Button: The input[type="submit"] button is styled with a green background, white text, and a hover effect that changes the button's background color.**
3. **Form Section:**
   * **Inside the <body>, the form is displayed within the .form-container.**
   * **The form uses the POST method and submits data to the same script (action set to <?php echo $\_SERVER['PHP\_SELF']; ?>).**
   * **The form includes two fields:**
     + **Name (text input, required).**
     + **Email (email input, required).**
   * **A submit button is provided to send the form data.**
4. **PHP Form Handling Section:**
   * **After form submission, the script checks if the form was submitted via the POST method (if ($\_SERVER["REQUEST\_METHOD"] == "POST")).**
   * **Data from the form is collected and sanitized using htmlspecialchars() to prevent XSS attacks.**
   * **The sanitized input is stored in two variables:**
     + **$name for the submitted name.**
     + **$email for the submitted email.**
   * **The script then displays a success message along with the processed data (name and email) using HTML.**

**3. Expected Output**

