|  |  |  |  |
| --- | --- | --- | --- |
| **Name:** | ORIHUELA, DANIELLE JOSHUA B. | **Date:** | 10/2/2023 |
| **Program:** | CPE3 | **Section:** | A361 |

**Intended Learning Outcomes:**

1. Use PHP scripts to receive, store, and manipulate user data.
2. Utilize PHP forms.

Create a form that has four input fields named Word1, Word2, Word3, and Word4 as well as “Reset” and “Submit” buttons. Create a form processing script that verifies that all four words are entered, that all of them contain only letters, and that all four words are between 4 and 7 characters long. Once all of the words have been verified as correct, use the strtoupper() and str\_shuffle() functions to produce four jumbled sets of letters.

The form must also have a distinct layout using HTML and CSS. You may opt to use a Bootstrap theme, but this is optional.

Here is a sample output:

**The Form:**

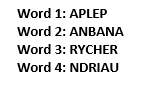
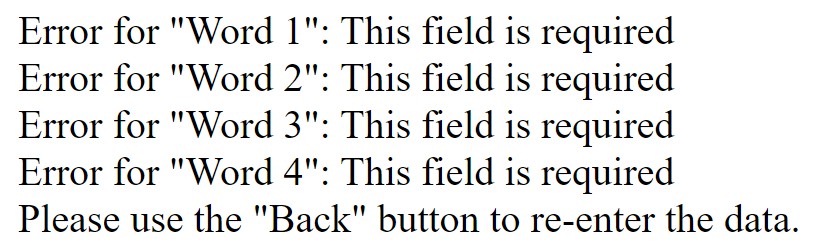
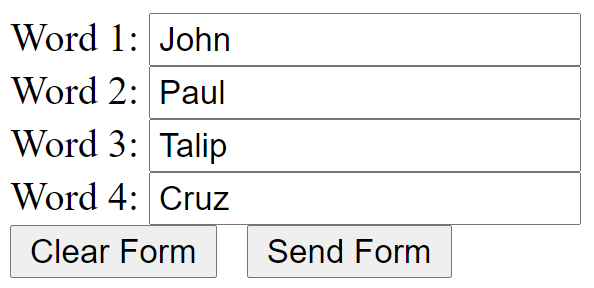
**APPLE BANANA CHERRY DURIAN**

**The Error Message:**

**The Output:**

Hint: Create functions that will display the error and validate the inputs of the user.

================================================================================



**HTML with PHP Source Code:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <link rel="stylesheet" href="LabTask9.css">

    <title>Word Jumbler</title>

</head>

<body>

    <div class="container">

        <h1>Word Jumbler</h1>

        <form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]); ?>">

            <div class="input-group">

                <label for="Word1">Word 1:</label>

                <input type="text" id="Word1" name="Word1" required>

            </div>

            <div class="input-group">

                <label for="Word2">Word 2:</label>

                <input type="text" id="Word2" name="Word2" required>

            </div>

            <div class="input-group">

                <label for="Word3">Word 3:</label>

                <input type="text" id="Word3" name="Word3" required>

            </div>

            <div class="input-group">

                <label for="Word4">Word 4:</label>

                <input type="text" id="Word4" name="Word4" required>

            </div>

            <div class="button-group">

                <input type="reset" value="Reset">

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

            </div>

        </form>

        <?php

$errors = array();

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

    $word1 = $\_POST["Word1"];

    $word2 = $\_POST["Word2"];

    $word3 = $\_POST["Word3"];

    $word4 = $\_POST["Word4"];

    // Validate words

    if (!preg\_match('/^[a-zA-Z]{4,7}$/', $word1)) {

        $errors[] = "Word 1 must contain only letters and be between 4 and 7 characters long.";

    }

    if (!preg\_match('/^[a-zA-Z]{4,7}$/', $word2)) {

        $errors[] = "Word 2 must contain only letters and be between 4 and 7 characters long.";

    }

    if (!preg\_match('/^[a-zA-Z]{4,7}$/', $word3)) {

        $errors[] = "Word 3 must contain only letters and be between 4 and 7 characters long.";

    }

    if (!preg\_match('/^[a-zA-Z]{4,7}$/', $word4)) {

        $errors[] = "Word 4 must contain only letters and be between 4 and 7 characters long.";

    }

    if (empty($errors)) {

        // Generate jumbled sets of letters

        $jumbled1 = str\_shuffle(strtoupper($word1));

        $jumbled2 = str\_shuffle(strtoupper($word2));

        $jumbled3 = str\_shuffle(strtoupper($word3));

        $jumbled4 = str\_shuffle(strtoupper($word4));

        echo "Word 1: $jumbled1<br>";

        echo "Word 2: $jumbled2<br>";

        echo "Word 3: $jumbled3<br>";

        echo "Word 4: $jumbled4<br>";

    } else {

        // Display error messages with added CSS classes

        foreach ($errors as $error) {

            echo '<div class="error-message">' . $error . '</div>';

        }

    }

}

?>

    </div>

</body>

</html>

**CSS Source Code:**

body {

    font-family: Arial, sans-serif;

    background-color: #f2f2f2;

}

.container {

    background-color: #ffffff;

    max-width: 400px;

    margin: 0 auto;

    padding: 20px;

    border-radius: 5px;

    box-shadow: 0 0 10px rgba(0, 0, 0, 0.2);

}

h1 {

    text-align: center;

    color: #333333;

}

.input-group {

    margin-bottom: 10px;

}

label {

    display: block;

    margin-bottom: 5px;

    color: #333333;

}

input[type="text"] {

    width: 100%;

    padding: 8px;

    border: 1px solid #ccc;

    border-radius: 3px;

    font-size: 16px;

}

.button-group {

    text-align: center;

}

input[type="reset"], input[type="submit"] {

    padding: 10px 20px;

    font-size: 16px;

    background-color: #007bff;

    color: #fff;

    border: none;

    border-radius: 3px;

    cursor: pointer;

}

input[type="reset"]:hover, input[type="submit"]:hover {

    background-color: #0056b3;

}

.error-message {

    color: #ff0000;

    font-size: 14px;

    margin-top: 5px;

}

.input-error {

    border: 1px solid #ff0000;

}

**Output**

A screenshot of a computer

Description automatically generated

**Findings and Observations:**

This labtask gave me a hard time doing because it has a lot of errors and I cannot find some solutions to it so I write codes again from scratch so that I can tract well the codes that I used.