FEU DILIMAN

DEPARTMENT OF COMPUTER STUDIES

(Applications Development and Emerging Technologies)

PRE-SUMMATIVE ASSESSMENT

5

USER INTERFACES, SESSION AND COOKIES

Student Name / Group Name:	CUETO, Alexa Joyce G.	
Members (if Group):	**Name**	**Role**
Section:	TW25	
Professor:	Sir Joseph Calleja	

I. PROGRAM OUTCOME/S (PO) ADDRESSED BY THE LABORATORY EXERCISE

- Design, implement and evaluate computer-based systems or applications to meet desired needs and requirements.

II. COURSE LEARNING OUTCOME/S (CLO) ADDRESSED BY THE LABORATORY EXERCISE

- Understand and apply best practices and standards in the development of website.

III. INTENDED LEARNING OUTCOME/S (ILO) OF THE LABORATORY EXERCISE

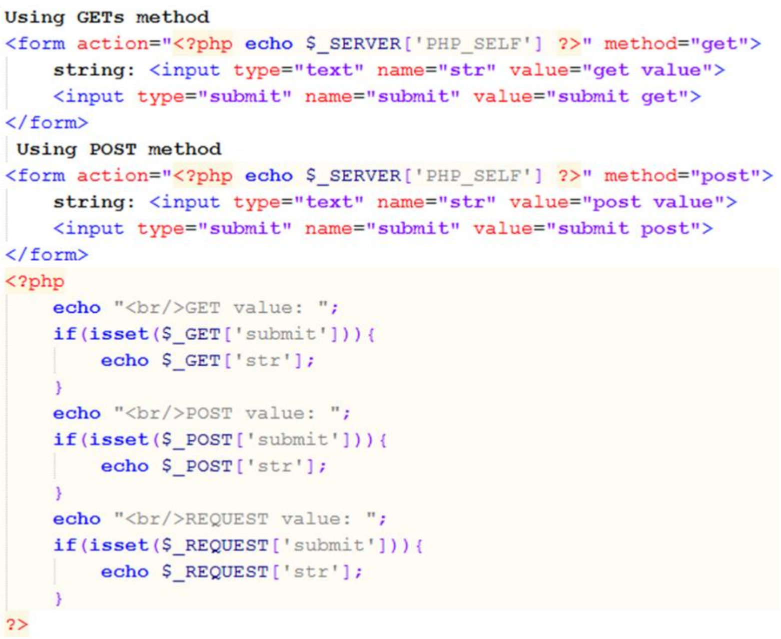
At the end of this exercise, students must be able to:

- To apply the available super global variables for form processing and validation.
- To differentiate the use of $_GET, $_POST, and $_REQUEST super global variable in form processing and know when to use it.
- To differentiate the use of Session and Cookies for form security of a Web Site.
- To know the proper syntax for validating user inputs using Regular Expression.

IV. BACKGROUND INFORMATION

Superglobal variables

Example:



```
Using GETs method
<form action="<?php echo $_SERVER['PHP_SELF'] ?>" method="get">
    string: <input type="text" name="str" value="get value">
    <input type="submit" name="submit" value="submit get">
</form>
 Using POST method
<form action="<?php echo $_SERVER['PHP_SELF'] ?>" method="post">
    string: <input type="text" name="str" value="post value">
    <input type="submit" name="submit" value="submit post">
</form>
<?php
    echo "<br/>GET value: ";
    if(isset($_GET['submit'])){
        echo $_GET['str'];
    }
    echo "<br/>POST value: ";
    if(isset($_POST['submit'])){
        echo $_POST['str'];
    }
    echo "<br/>REQUEST value: ";
    if(isset($_REQUEST['submit'])){
        echo $_REQUEST['str'];
    }
?>
```

PHP Superglobal - $GLOBALS

Super global variables are built-in variables that are always available in all scopes.

PHP $GLOBALS

$GLOBALS is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods).
PHP stores all global variables in an array called $GLOBALS[*index*].
The *index* holds the name of the variable.
The example below shows how to use the super global variable $GLOBALS:

Example

```
<?php
$x = 75;
$y = 25;

function addition() {
  $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];
}

addition();
echo $z;
?>
```

PHP Superglobal - $_POST

Super global variables are built-in variables that are always available in all scopes.

PHP $_POST

PHP $_POST is a PHP super global variable which is used to collect form data after submitting an HTML form with method="post". $_POST is also widely used to pass variables.

The example below shows a form with an input field and a submit button. When a user submits the data by clicking on "Submit", the form data is sent to the file specified in the action attribute of the <form> tag. In this example, we point to the file itself for processing form data. If you wish to use another PHP file to process form data, replace that with the filename of your choice. Then, we can use the super global variable $_POST to collect the value of the input field:

Example

```
<html>
<body>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  // collect value of input field
  $name = $_POST['fname'];
  if (empty($name)) {
    echo "Name is empty";
  } else {
    echo $name;
  }
}
?>

</body>
</html>
```

PHP Superglobal - $_GET

Super global variables are built-in variables that are always available in all scopes.

PHP $_GET

PHP $_GET is a PHP super global variable which is used to collect form data after submitting an HTML form with method="get".

$_GET can also collect data sent in the URL.

Assume we have an HTML page that contains a hyperlink with parameters:

```
<html>
<body>

<a href="test_get.php?subject=PHP&web=W3schools.com">Test $GET</a>
```

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

When a user clicks on the link "Test $GET", the parameters "subject" and "web" are sent to "test_get.php", and you can then access their values in "test_get.php" with $_GET.

The example below shows the code in "test_get.php":

Example

```
<html>
<body>

<?php
echo "Study " . $_GET['subject'] . " at " . $_GET['web'];
?>

</body>
</html>
```

PHP Superglobal - $_REQUEST

Super global variables are built-in variables that are always available in all scopes.

PHP $_REQUEST is a PHP super global variable which is used to collect data after submitting an HTML form.

The example below shows a form with an input field and a submit button. When a user submits the data by clicking on "Submit", the form data is sent to the file specified in the action attribute of the <form> tag. In this example, we point to this file itself for processing form data. If you wish to use another PHP file to process form data, replace that with the filename of your choice. Then, we can use the super global variable $_REQUEST to collect the value of the input field:

Example

```
<html>
<body>

<form method="post" action="<?php echo $_SERVER['PHP_SELF'];?>">
```

```
  Name: <input type="text" name="fname">
  <input type="submit">
</form>

<?php
if ($_SERVER["REQUEST_METHOD"] == "POST") {
  // collect value of input field
  $name = $_REQUEST['fname'];
  if (empty($name)) {
    echo "Name is empty";
  } else {
    echo $name;
  }
}
?>

</body>
</html>
```

PHP Cookies

A cookie is often used to identify a user. A cookie is a small file that the server embeds on the user's computer. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.

Create Cookies With PHP

A cookie is created with the `setcookie()` function.

Syntax

```
setcookie(name, value, expire, path, domain, secure, httponly);
```

Only the *name* parameter is required. All other parameters are optional.

PHP Create/Retrieve a Cookie

The following example creates a cookie named "user" with the value "John Doe". The cookie will expire after 30 days (86400 * 30). The "/" means that the cookie is available in entire website (otherwise, select the directory you prefer).

We then retrieve the value of the cookie "user" (using the global variable $_COOKIE). We also use the `isset()` function to find out if the cookie is set:

Example

```
<?php
$cookie_name = "user";
$cookie_value = "John Doe";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); //
86400 = 1 day
?>
<html>
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
  echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
  echo "Cookie '" . $cookie_name . "' is set!<br>";
  echo "Value is: " . $_COOKIE[$cookie_name];
}
?>

</body>
</html>
```

Note: The `setcookie()` function must appear BEFORE the <html> tag.

Note: The value of the cookie is automatically URLencoded when sending the cookie, and automatically decoded when received (to prevent URLencoding, use `setrawcookie()` instead).

Modify a Cookie Value

To modify a cookie, just set (again) the cookie using the `setcookie()` function:

Example

```
<?php
$cookie_name = "user";
$cookie_value = "Alex Porter";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/");
?>
<html>
```

```
<body>

<?php
if(!isset($_COOKIE[$cookie_name])) {
  echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
  echo "Cookie '" . $cookie_name . "' is set!<br>";
  echo "Value is: " . $_COOKIE[$cookie_name];
}
?>

</body>
</html>
```

Delete a Cookie

To delete a cookie, use the `setcookie()` function with an expiration date in the past:

Example

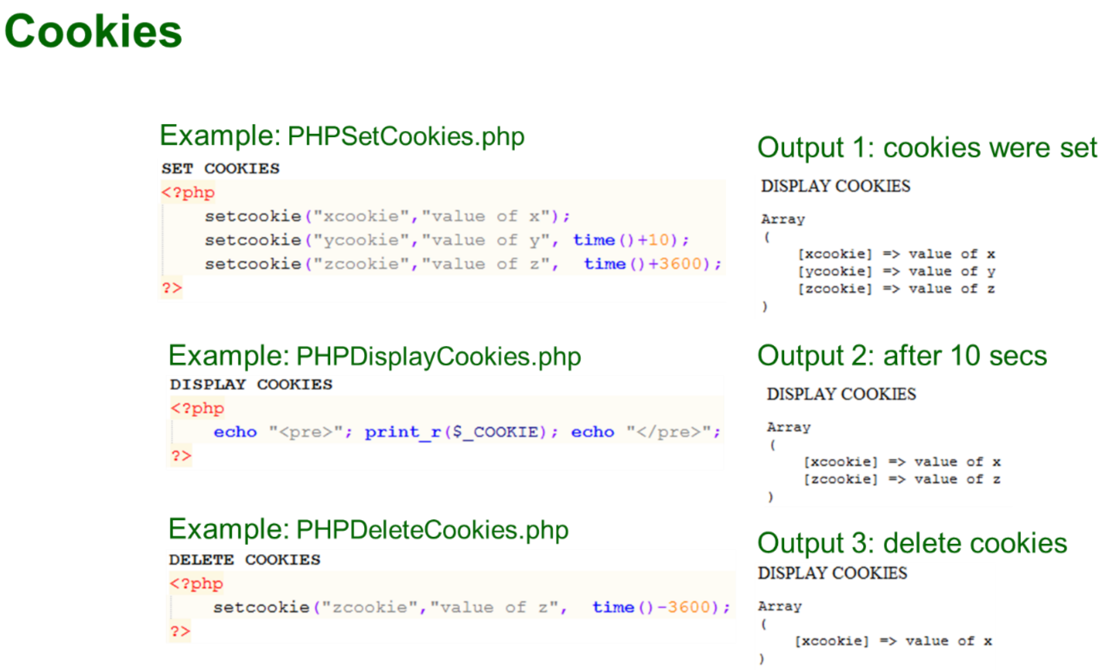
```
<?php
// set the expiration date to one hour ago
setcookie("user", "", time() - 3600);
?>
<html>
<body>

<?php
echo "Cookie 'user' is deleted.";
?>

</body>
</html>
```

Check if Cookies are Enabled

The following example creates a small script that checks whether cookies are enabled. First, try to create a test cookie with the `setcookie()` function, then count the $_COOKIE array variable:

Example

```
<?php
setcookie("test_cookie", "test", time() + 3600, '/');
?>
<html>
<body>

<?php
if(count($_COOKIE) > 0) {
  echo "Cookies are enabled.";
} else {
  echo "Cookies are disabled.";
}
?>

</body>
</html>
```



Cookies

Example: PHPSetCookies.php

```
SET COOKIES
<?php
    setcookie("xcookie","value of x");
    setcookie("ycookie","value of y", time()+10);
    setcookie("zcookie","value of z",  time()+3600);
?>
```

Output 1: cookies were set

```
DISPLAY COOKIES

Array
(
    [xcookie] => value of x
    [ycookie] => value of y
    [zcookie] => value of z
)
```

Example: PHPDisplayCookies.php

```
DISPLAY COOKIES
<?php
    echo "<pre>"; print_r($_COOKIE); echo "</pre>";
?>
```

Output 2: after 10 secs

```
DISPLAY COOKIES

Array
(
    [xcookie] => value of x
    [zcookie] => value of z
)
```

Example: PHPDeleteCookies.php

```
DELETE COOKIES
<?php
    setcookie("zcookie","value of z",  time()-3600);
?>
```

Output 3: delete cookies

```
DISPLAY COOKIES

Array
(
    [xcookie] => value of x
)
```

PHP Sessions

A session is a way to store information (in variables) to be used across multiple pages.

Unlike a cookie, the information is not stored on the users computer.

What is a PHP Session?

When you work with an application, you open it, do some changes, and then you close it. This is much like a Session. The computer knows who you are. It knows when you start the application and when you end. But on the internet there is one problem: the web server does not know who you are or what you do, because the HTTP address doesn't maintain state.

Session variables solve this problem by storing user information to be used across multiple pages (e.g. username, favorite color, etc). By default, session variables last until the user closes the browser.

So; Session variables hold information about one single user, and are available to all pages in one application.

Tip: If you need a permanent storage, you may want to store the data in a database.

Start a PHP Session

A session is started with the `session_start()` function.

Session variables are set with the PHP global variable: $_SESSION.

Now, let's create a new page called "demo_session1.php". In this page, we start a new PHP session and set some session variables:

Example

```
<?php
// Start the session
session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
// Set session variables
$_SESSION["favcolor"] = "green";
$_SESSION["favanimal"] = "cat";
echo "Session variables are set.";
```

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

Note: The `session_start()` function must be the very first thing in your document. Before any HTML tags.

Get PHP Session Variable Values

Next, we create another page called "demo_session2.php". From this page, we will access the session information we set on the first page ("demo_session1.php").

Notice that session variables are not passed individually to each new page, instead they are retrieved from the session we open at the beginning of each page (`session_start()`).

Also notice that all session variable values are stored in the global $_SESSION variable:

Example

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
// Echo session variables that were set on previous page
echo "Favorite color is " . $_SESSION["favcolor"] . ".<br>";
echo "Favorite animal is " . $_SESSION["favanimal"] . ".";
?>
</body>
</html>
```

Another way to show all the session variable values for a user session is to run the following code:

Example

```
<?php
session_start();
?>
```

```
<!DOCTYPE html>
<html>
<body>
<?php
print_r($_SESSION);
?>
</body>
</html>
```

Modify a PHP Session Variable

To change a session variable, just overwrite it:

Example

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>

<?php
// to change a session variable, just overwrite it
$_SESSION["favcolor"] = "yellow";
print_r($_SESSION);
?>
</body>
</html>
```

Destroy a PHP Session

To remove all global session variables and destroy the session, use `session_unset()` and `session_destroy()`:

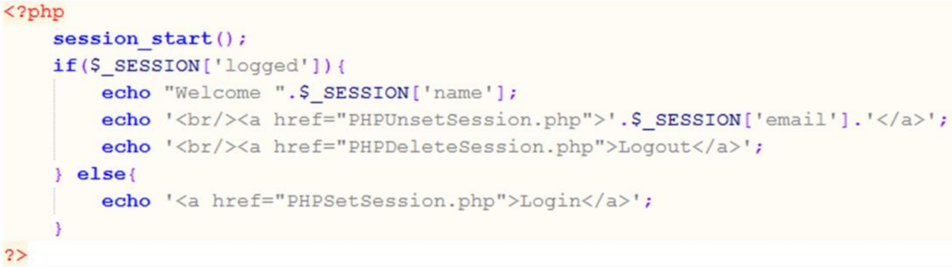
Example

```
<?php
session_start();
?>
<!DOCTYPE html>
<html>
<body>
<?php
```

```
// remove all session variables
session_unset();
// destroy the session
session_destroy();
?>
</body>
</html>
```

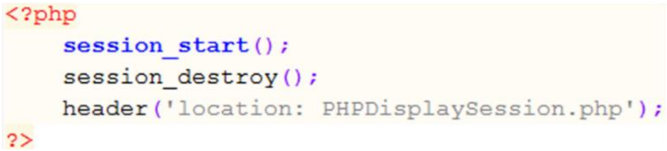
Session

Example: PHPDisplaySession.php



```
<?php
    session_start();
    if($_SESSION['logged']){
        echo "Welcome ".$_SESSION['name'];
        echo '<br/><a href="PHPUnsetSession.php">'.$_SESSION['email'].'</a>';
        echo '<br/><a href="PHPDeleteSession.php">Logout</a>';
    } else{
        echo '<a href="PHPSetSession.php">Login</a>';
    }
?>
```

Example: PHPDeleteSession.php

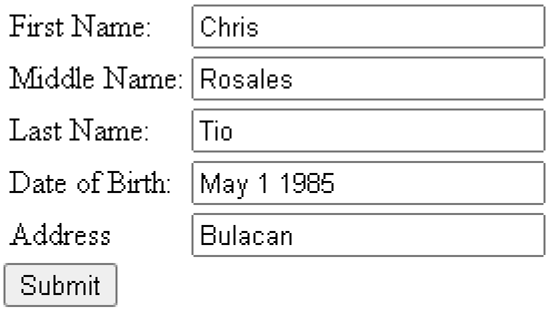


```
<?php
    session_start();
    session_destroy();
    header('location: PHPDisplaySession.php');
?>
```

V. GRADING SYSTEM / RUBRIC (please see separate sheet)

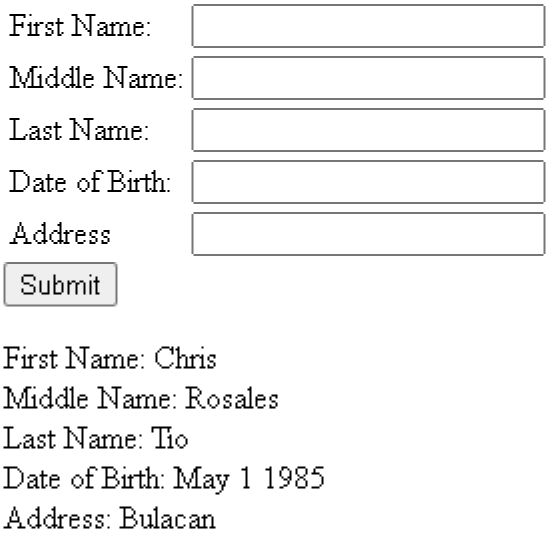
VI. LABORATORY ACTIVITY

1. Create a personal information webpage. Using $_GET and $_POST get the data (firstname, middlename, lastname, date of birth and address) from the forms and display below.



First Name: Chris
Middle Name: Rosales
Last Name: Tio
Date of Birth: May 1 1985
Address Bulacan

After click the submit button:


First Name:

Middle Name:

Last Name:

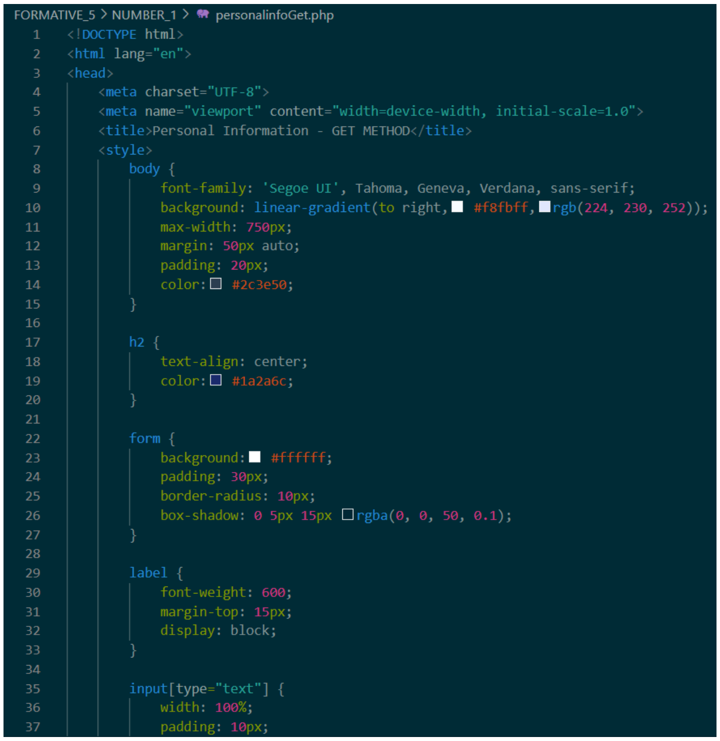
Date of Birth:

Address

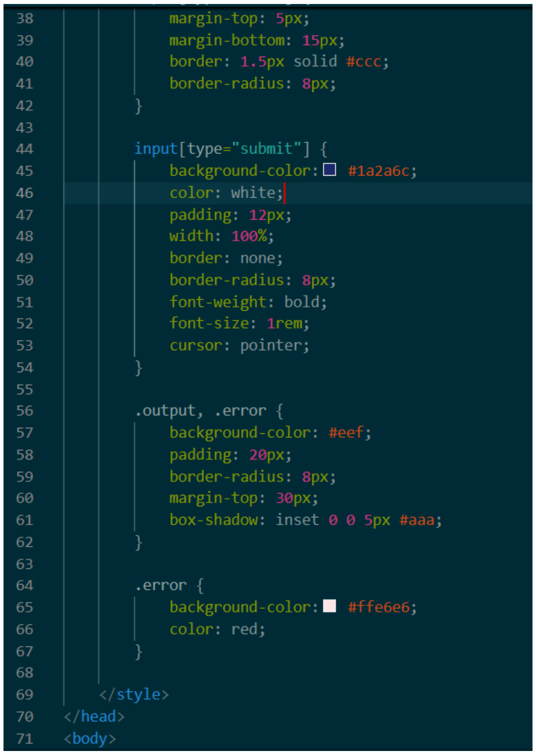
```
First Name: Chris
Middle Name: Rosales
Last Name: Tio
Date of Birth: May 1 1985
Address: Bulacan
```

Note: you need to create 2 php webpages, 1 is for $_GET function and the other 1 is for $_POST function.

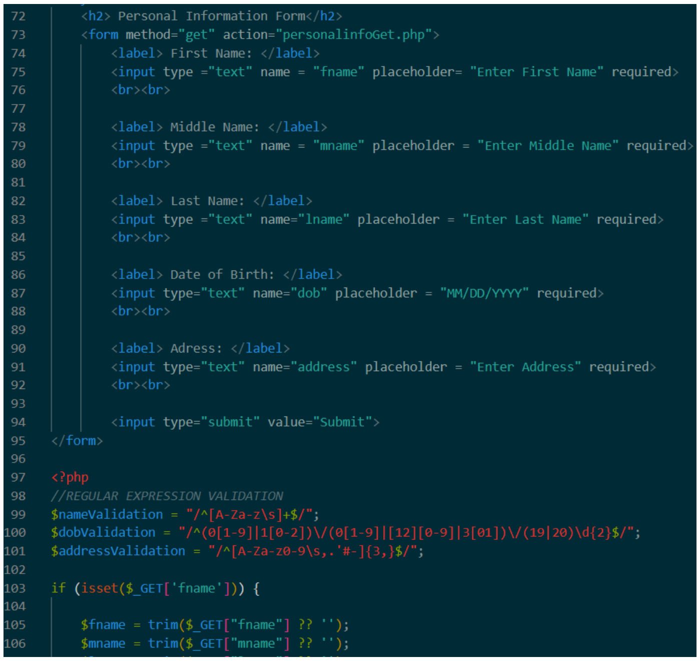
SOURCE CODE ($_GET FUNCTION)


FORMATIVE_5 > NUMBER_1 > personalinfoGet.php

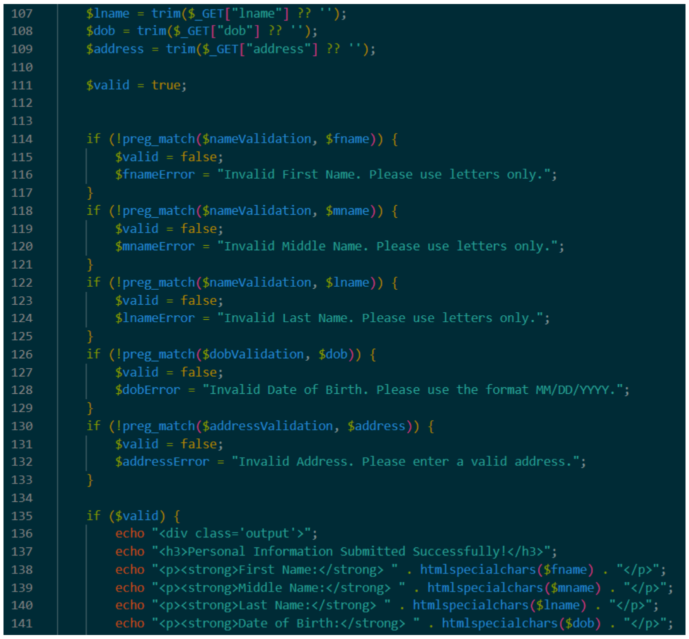
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Personal Information - GET METHOD</title>
    <style>
        body {
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
            background: linear-gradient(to right, #f8fbff, rgb(224, 230, 252));
            max-width: 750px;
            margin: 50px auto;
            padding: 20px;
            color: #2c3e50;
        }

        h2 {
            text-align: center;
            color: #1a2a6c;
        }

        form {
            background: #ffffff;
            padding: 30px;
            border-radius: 10px;
            box-shadow: 0 5px 15px rgba(0, 0, 50, 0.1);
        }

        label {
            font-weight: 600;
            margin-top: 15px;
            display: block;
        }

        input[type="text"] {
            width: 100%;
            padding: 10px;
```



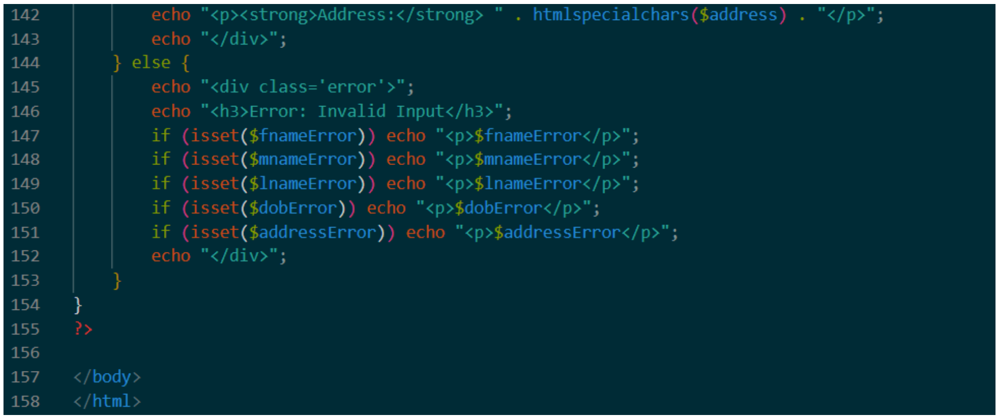
```
            margin-top: 5px;
            margin-bottom: 15px;
            border: 1.5px solid #ccc;
            border-radius: 8px;
        }

        input[type="submit"] {
            background-color: #1a2a6c;
            color: white;
            padding: 12px;
            width: 100%;
            border: none;
            border-radius: 8px;
            font-weight: bold;
            font-size: 1rem;
            cursor: pointer;
        }

        .output, .error {
            background-color: #eef;
            padding: 20px;
            border-radius: 8px;
            margin-top: 30px;
            box-shadow: inset 0 0 5px #aaa;
        }

        .error {
            background-color: #ffe6e6;
            color: red;
        }

    </style>
</head>
<body>
```



```
    <h2> Personal Information Form</h2>
    <form method="get" action="personalinfoGet.php">
        <label> First Name: </label>
        <input type ="text" name = "fname" placeholder= "Enter First Name" required>
        <br><br>

        <label> Middle Name: </label>
        <input type ="text" name = "mname" placeholder = "Enter Middle Name" required>
        <br><br>

        <label> Last Name: </label>
        <input type ="text" name="lname" placeholder = "Enter Last Name" required>
        <br><br>

        <label> Date of Birth: </label>
        <input type="text" name="dob" placeholder = "MM/DD/YYYY" required>
        <br><br>

        <label> Adress: </label>
        <input type="text" name="address" placeholder = "Enter Address" required>
        <br><br>

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

<?php
//REGULAR EXPRESSION VALIDATION
$nameValidation = "/^[A-Za-z\s]+$/";
$dobValidation = "/^(0[1-9]|1[0-2])\/(0[1-9]|[12][0-9]|3[01])\/(19|20)\d{2}$/";
$addressValidation = "/^[A-Za-z0-9\s,.'#-]{3,}$/";

if (isset($_GET['fname'])) {

    $fname = trim($_GET["fname"] ?? '');
    $mname = trim($_GET["mname"] ?? '');
```

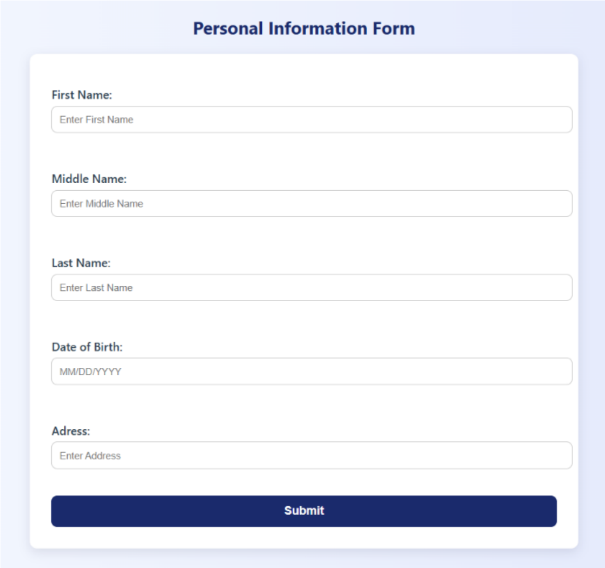


```
    $lname = trim($_GET["lname"] ?? '');
    $dob = trim($_GET["dob"] ?? '');
    $address = trim($_GET["address"] ?? '');

    $valid = true;


    if (!preg_match($nameValidation, $fname)) {
        $valid = false;
        $fnameError = "Invalid First Name. Please use letters only.";
    }
    if (!preg_match($nameValidation, $mname)) {
        $valid = false;
        $mnameError = "Invalid Middle Name. Please use letters only.";
    }
    if (!preg_match($nameValidation, $lname)) {
        $valid = false;
        $lnameError = "Invalid Last Name. Please use letters only.";
    }
    if (!preg_match($dobValidation, $dob)) {
        $valid = false;
        $dobError = "Invalid Date of Birth. Please use the format MM/DD/YYYY.";
    }
    if (!preg_match($addressValidation, $address)) {
        $valid = false;
        $addressError = "Invalid Address. Please enter a valid address.";
    }

    if ($valid) {
        echo "<div class='output'>";
        echo "<h3>Personal Information Submitted Successfully!</h3>";
        echo "<p><strong>First Name:</strong> " . htmlspecialchars($fname) . "</p>";
        echo "<p><strong>Middle Name:</strong> " . htmlspecialchars($mname) . "</p>";
        echo "<p><strong>Last Name:</strong> " . htmlspecialchars($lname) . "</p>";
        echo "<p><strong>Date of Birth:</strong> " . htmlspecialchars($dob) . "</p>";
```



```
        echo "<p><strong>Address:</strong> " . htmlspecialchars($address) . "</p>";
        echo "</div>";
    } else {
        echo "<div class='error'>";
        echo "<h3>Error: Invalid Input</h3>";
        if (isset($fnameError)) echo "<p>$fnameError</p>";
        if (isset($mnameError)) echo "<p>$mnameError</p>";
        if (isset($lnameError)) echo "<p>$lnameError</p>";
        if (isset($dobError)) echo "<p>$dobError</p>";
        if (isset($addressError)) echo "<p>$addressError</p>";
        echo "</div>";
    }
}
?>

</body>
</html>
```

OUTPUT:



Personal Information Form

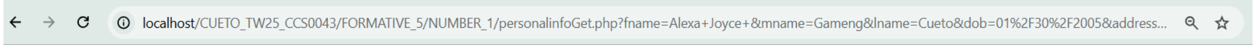
First Name:
Enter First Name

Middle Name:
Enter Middle Name

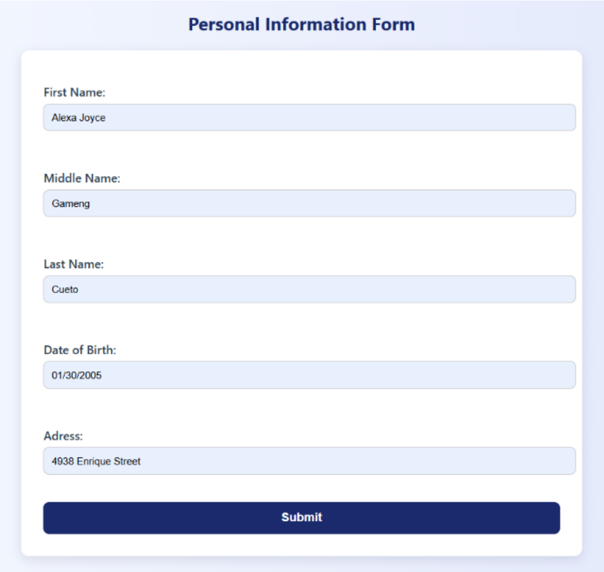
Last Name:
Enter Last Name

Date of Birth:
MM/DD/YYYY

Adress:
Enter Address



localhost/CUETO_TW25_CCS0043/FORMATIVE_5/NUMBER_1/personalinfoGet.php?fname=Alexa+Joyce+&mname=Gameng&lname=Cueto&dob=01%2F30%2F2005&address...



Personal Information Form

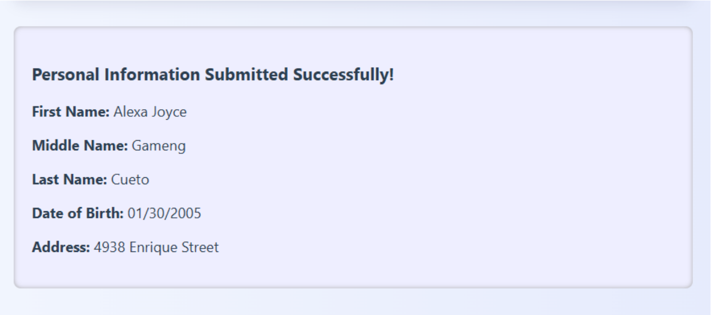
First Name: Alexa Joyce

Middle Name: Gameng

Last Name: Cueto

Date of Birth: 01/30/2005

Adress: 4938 Enrique Street



Personal Information Submitted Successfully!

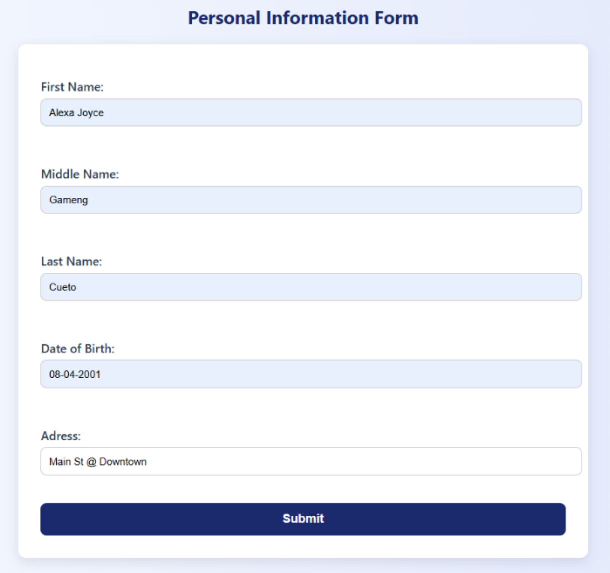
First Name: Alexa Joyce

Middle Name: Gameng

Last Name: Cueto

Date of Birth: 01/30/2005

Address: 4938 Enrique Street



Personal Information Form

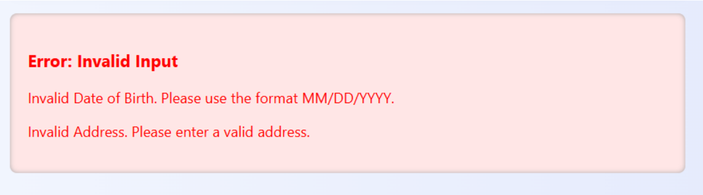
First Name: Alexa Joyce

Middle Name: Gameng

Last Name: Cueto

Date of Birth: 08-04-2001

Adress: Main St @ Downtown

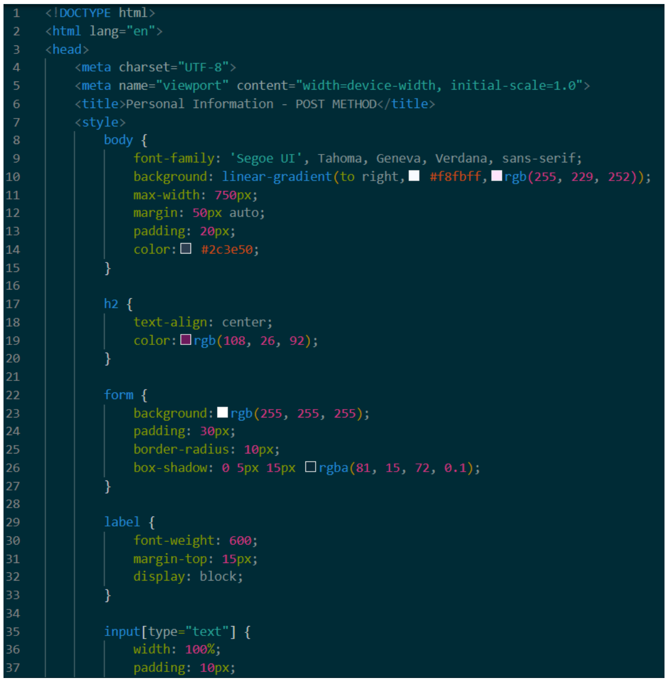


Error: Invalid Input

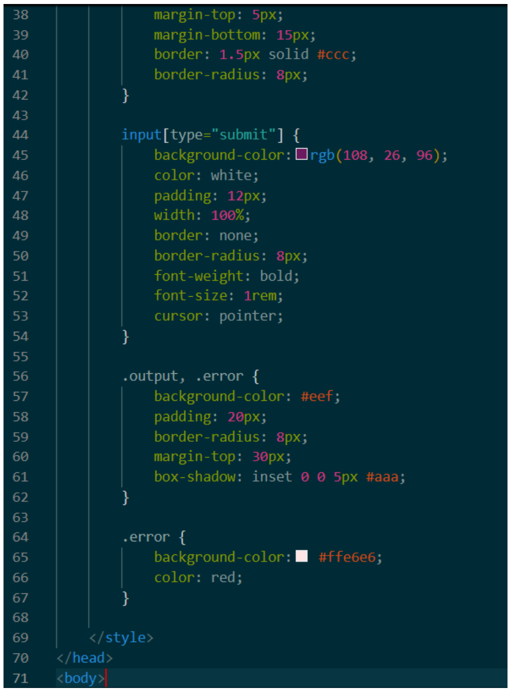
Invalid Date of Birth. Please use the format MM/DD/YYYY.

Invalid Address. Please enter a valid address.

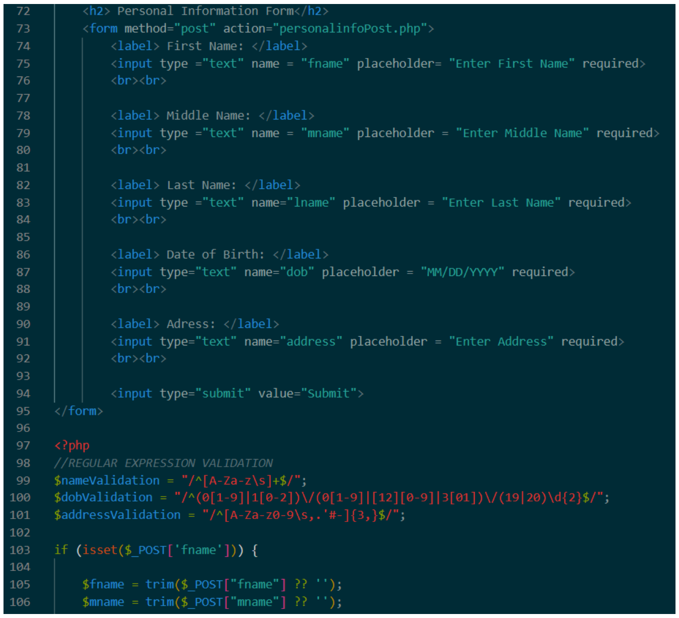
SOURCE CODE ($_POST FUNCTION)



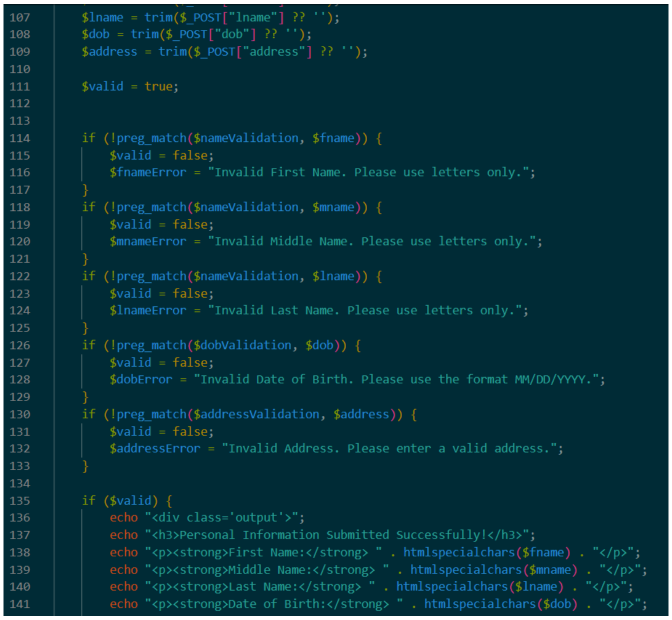
```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Personal Information - POST METHOD</title>
    <style>
        body {
            font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
            background: linear-gradient(to right, #f8fbff, rgb(255, 229, 252));
            max-width: 750px;
            margin: 50px auto;
            padding: 20px;
            color: #2c3e50;
        }

        h2 {
            text-align: center;
            color: rgb(108, 26, 92);
        }

        form {
            background: rgb(255, 255, 255);
            padding: 30px;
            border-radius: 10px;
            box-shadow: 0 5px 15px rgba(81, 15, 72, 0.1);
        }

        label {
            font-weight: 600;
            margin-top: 15px;
            display: block;
        }

        input[type="text"] {
            width: 100%;
            padding: 10px;
```



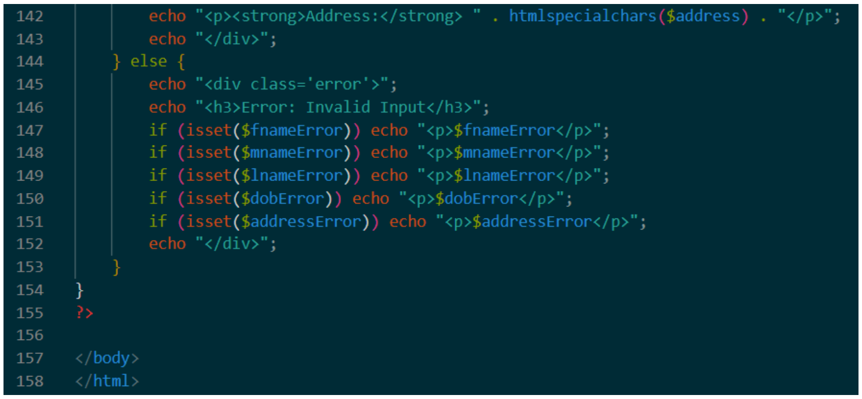
```html
            margin-top: 5px;
            margin-bottom: 15px;
            border: 1.5px solid #ccc;
            border-radius: 8px;
        }

        input[type="submit"] {
            background-color: rgb(108, 26, 96);
            color: white;
            padding: 12px;
            width: 100%;
            border: none;
            border-radius: 8px;
            font-weight: bold;
            font-size: 1rem;
            cursor: pointer;
        }

        .output, .error {
            background-color: #eef;
            padding: 20px;
            border-radius: 8px;
            margin-top: 30px;
            box-shadow: inset 0 0 5px #aaa;
        }

        .error {
            background-color: #ffe6e6;
            color: red;
        }

    </style>
</head>
<body>
```



```php
    <h2> Personal Information Form</h2>
    <form method="post" action="personalinfoPost.php">
        <label> First Name: </label>
        <input type ="text" name = "fname" placeholder= "Enter First Name" required>
        <br><br>

        <label> Middle Name: </label>
        <input type ="text" name = "mname" placeholder = "Enter Middle Name" required>
        <br><br>

        <label> Last Name: </label>
        <input type ="text" name="lname" placeholder = "Enter Last Name" required>
        <br><br>

        <label> Date of Birth: </label>
        <input type="text" name="dob" placeholder = "MM/DD/YYYY" required>
        <br><br>

        <label> Adress: </label>
        <input type="text" name="address" placeholder = "Enter Address" required>
        <br><br>

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

<?php
//REGULAR EXPRESSION VALIDATION
$nameValidation = "/^[A-Za-z\s]+$/";
$dobValidation = "/^(0[1-9]|1[0-2])\/(0[1-9]|[12][0-9]|3[01])\/(19|20)\d{2}$/";
$addressValidation = "/^[A-Za-z0-9\s,.'#-]{3,}$/";

if (isset($_POST['fname'])) {

    $fname = trim($_POST["fname"] ?? '');
    $mname = trim($_POST["mname"] ?? '');
```

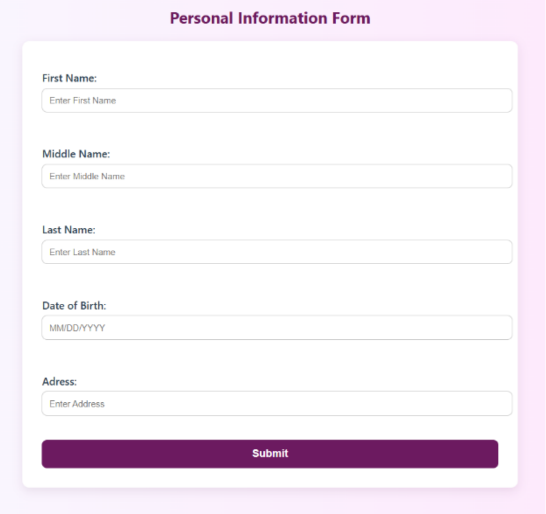


```php
    $lname = trim($_POST["lname"] ?? '');
    $dob = trim($_POST["dob"] ?? '');
    $address = trim($_POST["address"] ?? '');

    $valid = true;


    if (!preg_match($nameValidation, $fname)) {
        $valid = false;
        $fnameError = "Invalid First Name. Please use letters only.";
    }
    if (!preg_match($nameValidation, $mname)) {
        $valid = false;
        $mnameError = "Invalid Middle Name. Please use letters only.";
    }
    if (!preg_match($nameValidation, $lname)) {
        $valid = false;
        $lnameError = "Invalid Last Name. Please use letters only.";
    }
    if (!preg_match($dobValidation, $dob)) {
        $valid = false;
        $dobError = "Invalid Date of Birth. Please use the format MM/DD/YYYY.";
    }
    if (!preg_match($addressValidation, $address)) {
        $valid = false;
        $addressError = "Invalid Address. Please enter a valid address.";
    }

    if ($valid) {
        echo "<div class='output'>";
        echo "<h3>Personal Information Submitted Successfully!</h3>";
        echo "<p><strong>First Name:</strong> " . htmlspecialchars($fname) . "</p>";
        echo "<p><strong>Middle Name:</strong> " . htmlspecialchars($mname) . "</p>";
        echo "<p><strong>Last Name:</strong> " . htmlspecialchars($lname) . "</p>";
        echo "<p><strong>Date of Birth:</strong> " . htmlspecialchars($dob) . "</p>";
```



```
        echo "<p><strong>Address:</strong> " . htmlspecialchars($address) . "</p>";
        echo "</div>";
    } else {
        echo "<div class='error'>";
        echo "<h3>Error: Invalid Input</h3>";
        if (isset($fnameError)) echo "<p>$fnameError</p>";
        if (isset($mnameError)) echo "<p>$mnameError</p>";
        if (isset($lnameError)) echo "<p>$lnameError</p>";
        if (isset($dobError)) echo "<p>$dobError</p>";
        if (isset($addressError)) echo "<p>$addressError</p>";
        echo "</div>";
    }
}
?>

</body>
</html>
```

OUTPUT:



Personal Information Form

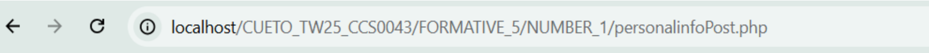
First Name:
Enter First Name

Middle Name:
Enter Middle Name

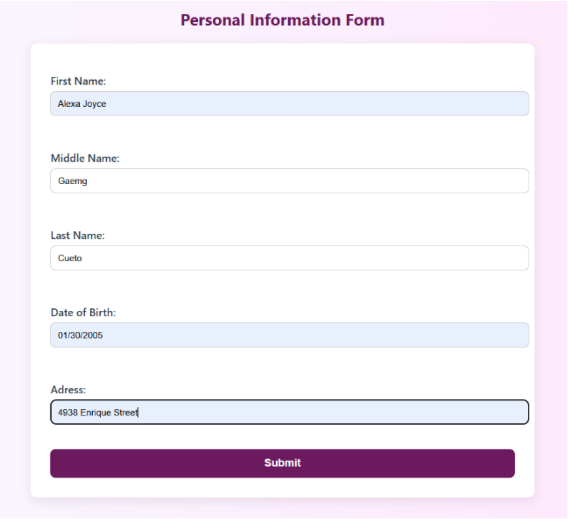
Last Name:
Enter Last Name

Date of Birth:
MM/DD/YYYY

Adress:
Enter Address



localhost/CUETO_TW25_CCS0043/FORMATIVE_5/NUMBER_1/personalinfoPost.php



Personal Information Form

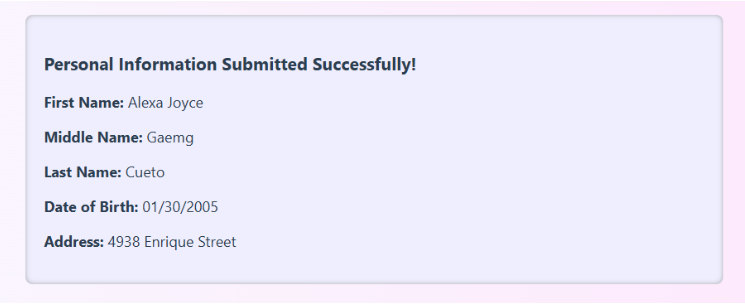
First Name:
Alexa Joyce

Middle Name:
Gaemg

Last Name:
Cueto

Date of Birth:
01/30/2005

Adress:
4938 Enrique Street



Personal Information Submitted Successfully!

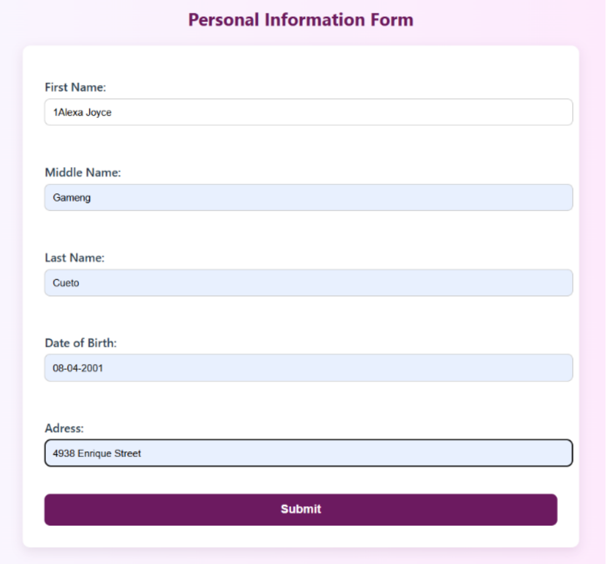
First Name: Alexa Joyce

Middle Name: Gaemg

Last Name: Cueto

Date of Birth: 01/30/2005

Address: 4938 Enrique Street



Personal Information Form

First Name:

1Alexa Joyce

Middle Name:

Gameng

Last Name:

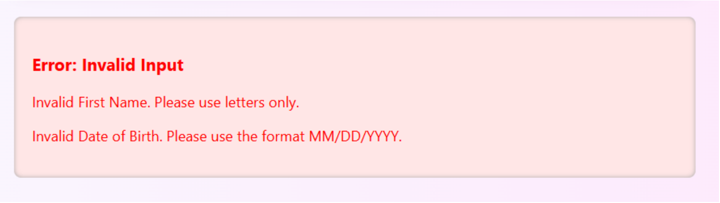
Cueto

Date of Birth:

08-04-2001

Adress:

4938 Enrique Street



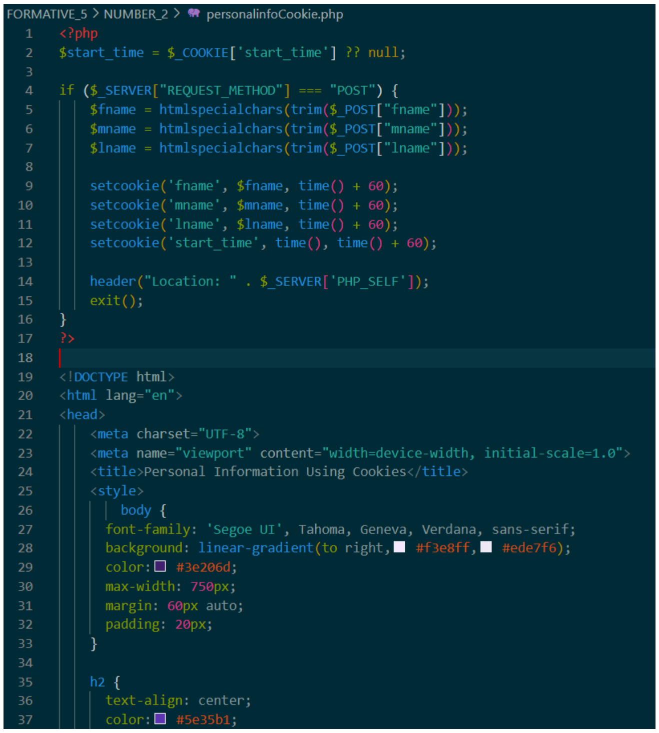
Error: Invalid Input

Invalid First Name. Please use letters only.

Invalid Date of Birth. Please use the format MM/DD/YYYY.

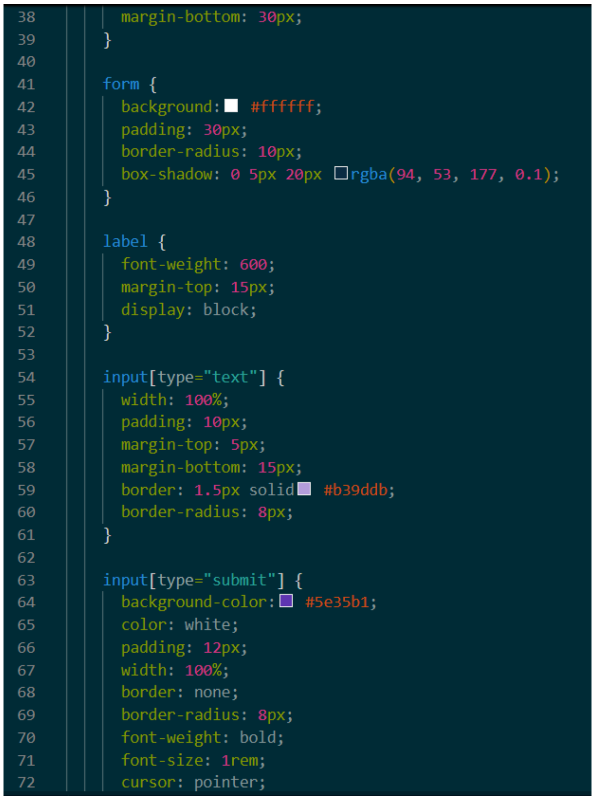
2. Create a personal information webpage. Using setcookie() function, create cookies for the firstname, middle name and lastname. It must be show the cookies after 10, 20 and 30 seconds.

SOURCE CODE:

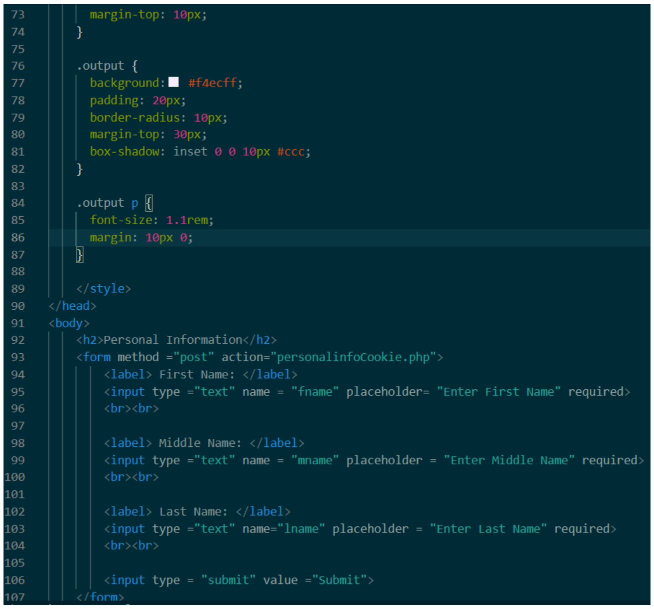


FORMATIVE_5 > NUMBER_2 > personalinfoCookie.php

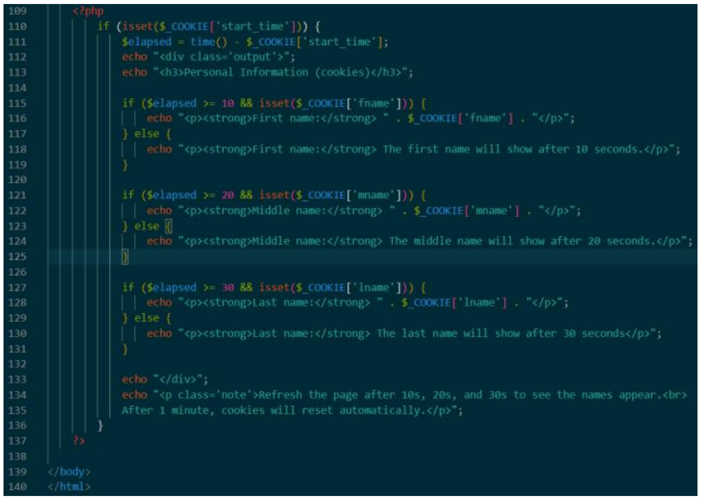
```php
<?php
$start_time = $_COOKIE['start_time'] ?? null;

if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $fname = htmlspecialchars(trim($_POST["fname"]));
    $mname = htmlspecialchars(trim($_POST["mname"]));
    $lname = htmlspecialchars(trim($_POST["lname"]));

    setcookie('fname', $fname, time() + 60);
    setcookie('mname', $mname, time() + 60);
    setcookie('lname', $lname, time() + 60);
    setcookie('start_time', time(), time() + 60);

    header("Location: " . $_SERVER['PHP_SELF']);
    exit();
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Personal Information Using Cookies</title>
    <style>
        body {
      font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
      background: linear-gradient(to right, #f3e8ff, #ede7f6);
      color: #3e206d;
      max-width: 750px;
      margin: 60px auto;
      padding: 20px;
    }

    h2 {
      text-align: center;
      color: #5e35b1;
```



```css
      margin-bottom: 30px;
    }

    form {
      background: #ffffff;
      padding: 30px;
      border-radius: 10px;
      box-shadow: 0 5px 20px rgba(94, 53, 177, 0.1);
    }

    label {
      font-weight: 600;
      margin-top: 15px;
      display: block;
    }

    input[type="text"] {
      width: 100%;
      padding: 10px;
      margin-top: 5px;
      margin-bottom: 15px;
      border: 1.5px solid #b39ddb;
      border-radius: 8px;
    }

    input[type="submit"] {
      background-color: #5e35b1;
      color: white;
      padding: 12px;
      width: 100%;
      border: none;
      border-radius: 8px;
      font-weight: bold;
      font-size: 1rem;
      cursor: pointer;
```

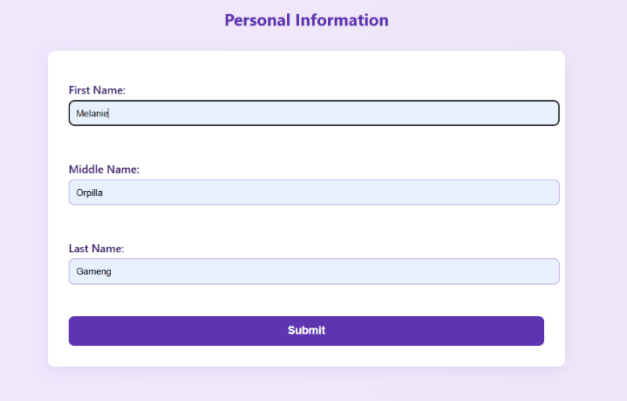


```
        margin-top: 10px;
      }

      .output {
        background: #f4ecff;
        padding: 20px;
        border-radius: 10px;
        margin-top: 30px;
        box-shadow: inset 0 0 10px #ccc;
      }

      .output p {
        font-size: 1.1rem;
        margin: 10px 0;
      }

      </style>
  </head>
  <body>
      <h2>Personal Information</h2>
      <form method ="post" action="personalinfoCookie.php">
          <label> First Name: </label>
          <input type ="text" name = "fname" placeholder= "Enter First Name" required>
          <br><br>

          <label> Middle Name: </label>
          <input type ="text" name = "mname" placeholder = "Enter Middle Name" required>
          <br><br>

          <label> Last Name: </label>
          <input type ="text" name="lname" placeholder = "Enter Last Name" required>
          <br><br>

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



```
      <?php
        if (isset($_COOKIE['start_time'])) {
            $elapsed = time() - $_COOKIE['start_time'];
            echo "<div class='output'>";
            echo "<h3>Personal Information (cookies)</h3>";

            if ($elapsed >= 10 && isset($_COOKIE['fname'])) {
                echo "<p><strong>First name:</strong> " . $_COOKIE['fname'] . "</p>";
            } else {
                echo "<p><strong>First name:</strong> The first name will show after 10 seconds.</p>";
            }

            if ($elapsed >= 20 && isset($_COOKIE['mname'])) {
                echo "<p><strong>Middle name:</strong> " . $_COOKIE['mname'] . "</p>";
            } else {
                echo "<p><strong>Middle name:</strong> The middle name will show after 20 seconds.</p>";
            }

            if ($elapsed >= 30 && isset($_COOKIE['lname'])) {
                echo "<p><strong>Last name:</strong> " . $_COOKIE['lname'] . "</p>";
            } else {
                echo "<p><strong>Last name:</strong> The last name will show after 30 seconds</p>";
            }

            echo "</div>";
            echo "<p class='note'>Refresh the page after 10s, 20s, and 30s to see the names appear.<br>
            After 1 minute, cookies will reset automatically.</p>";
        }
      ?>

  </body>
</html>
```

OUTPUT:

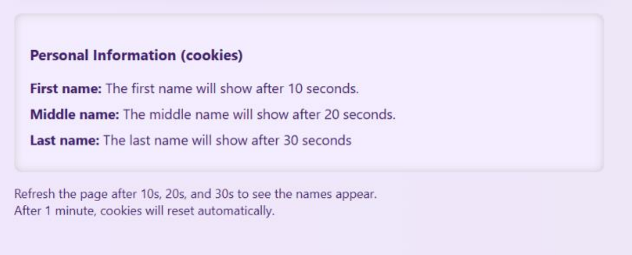


Personal Information

First Name: Melanie

Middle Name: Orpilla

Last Name: Gameng



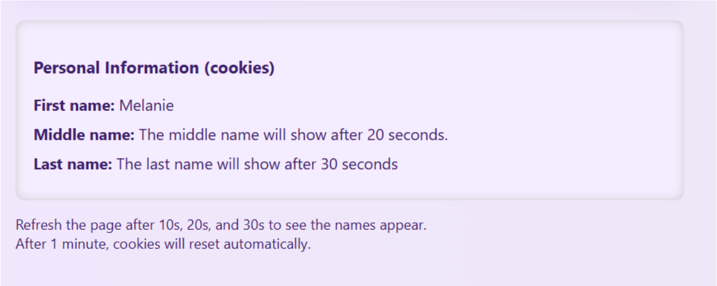
Personal Information (cookies)

First name: The first name will show after 10 seconds.

Middle name: The middle name will show after 20 seconds.

Last name: The last name will show after 30 seconds

Refresh the page after 10s, 20s, and 30s to see the names appear.
After 1 minute, cookies will reset automatically.



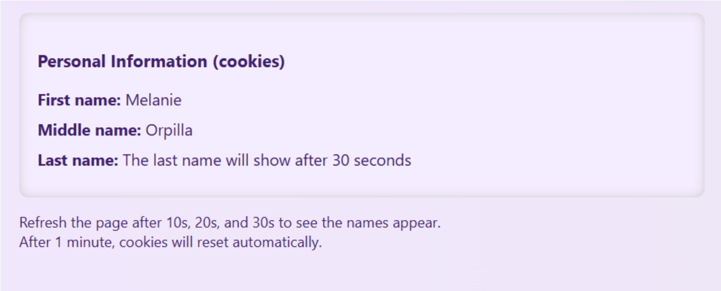
Personal Information (cookies)

First name: Melanie

Middle name: The middle name will show after 20 seconds.

Last name: The last name will show after 30 seconds

Refresh the page after 10s, 20s, and 30s to see the names appear.
After 1 minute, cookies will reset automatically.



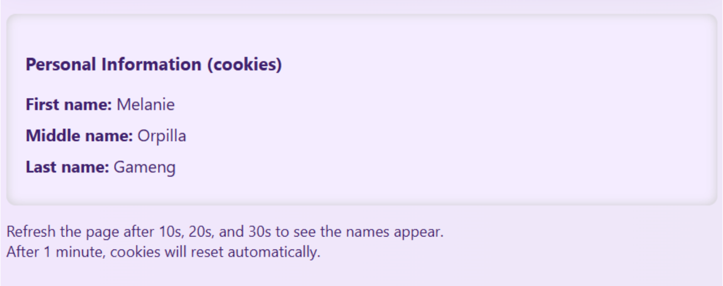
Personal Information (cookies)

First name: Melanie

Middle name: Orpilla

Last name: The last name will show after 30 seconds

Refresh the page after 10s, 20s, and 30s to see the names appear.
After 1 minute, cookies will reset automatically.



Personal Information (cookies)

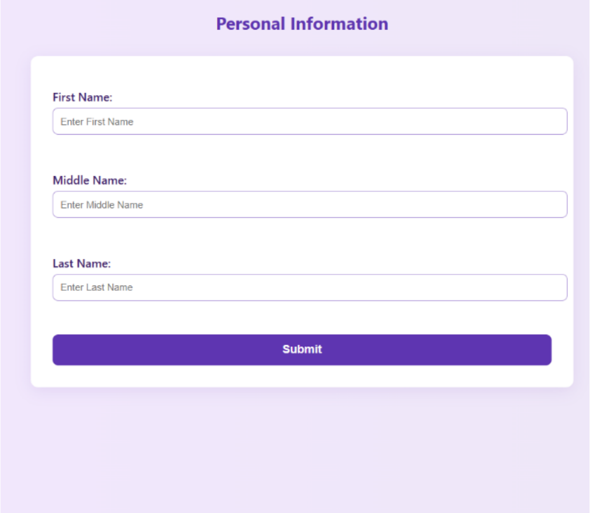
First name: Melanie

Middle name: Orpilla

Last name: Gameng

Refresh the page after 10s, 20s, and 30s to see the names appear.
After 1 minute, cookies will reset automatically.

After one minute:



Personal Information

First Name:

Enter First Name

Middle Name:

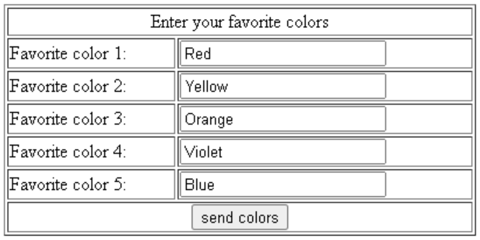
Enter Middle Name

Last Name:

Enter Last Name

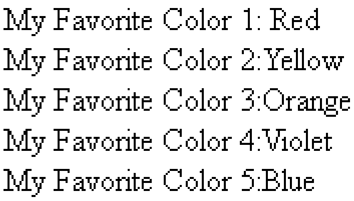
3. Create a webpage that will set a session for your 5 favorite colors and use it to the other webpage. Integrate HTML and CSS, you can create your own design and include images. Please see the example below:

FavoriteColor.php



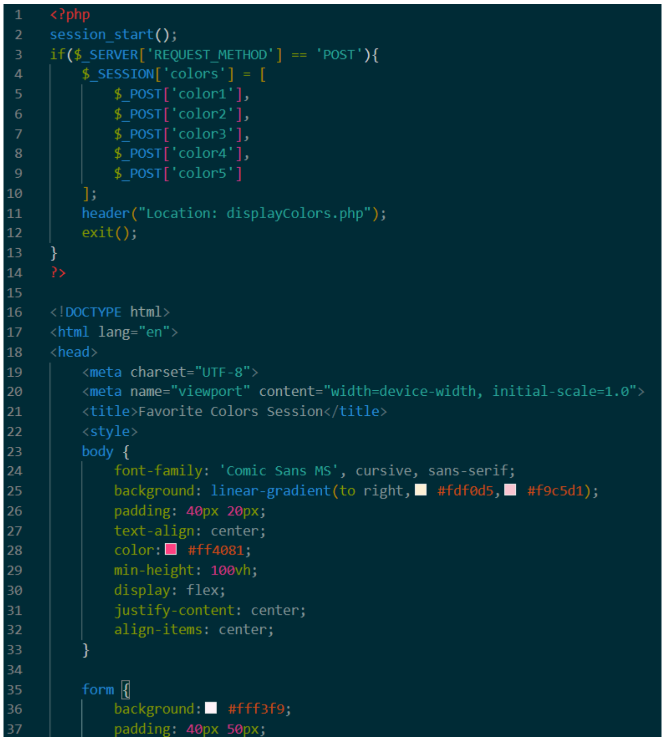
Enter your favorite colors	
Favorite color 1:	Red
Favorite color 2:	Yellow
Favorite color 3:	Orange
Favorite color 4:	Violet
Favorite color 5:	Blue
send colors	

After click the button send colors:
ResultColors.php

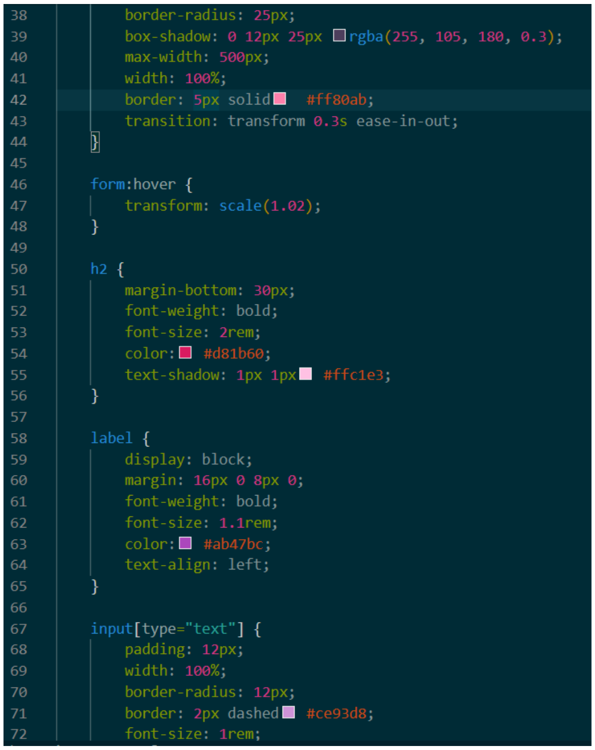


My Favorite Color 1: Red
My Favorite Color 2:Yellow
My Favorite Color 3:Orange
My Favorite Color 4:Violet
My Favorite Color 5:Blue

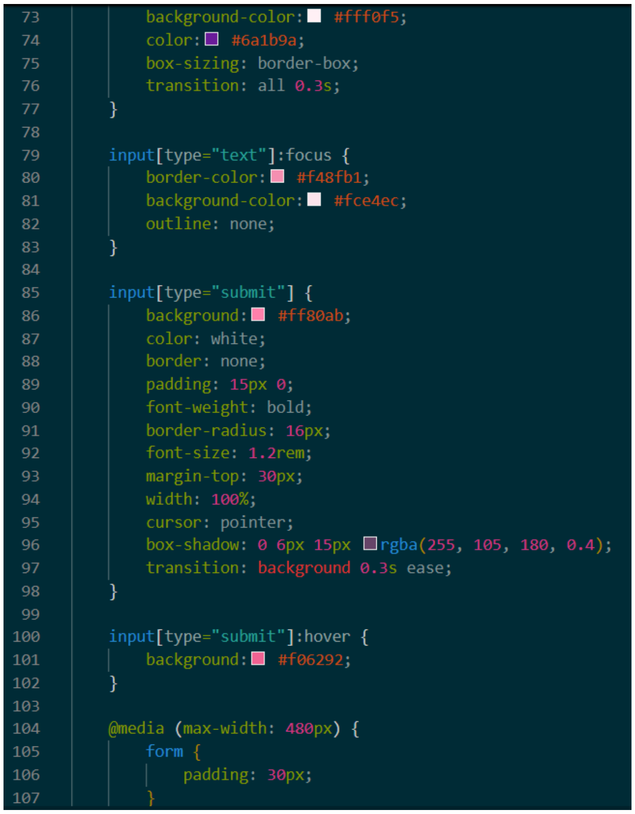
SOURCE CODE (colorSession.php)



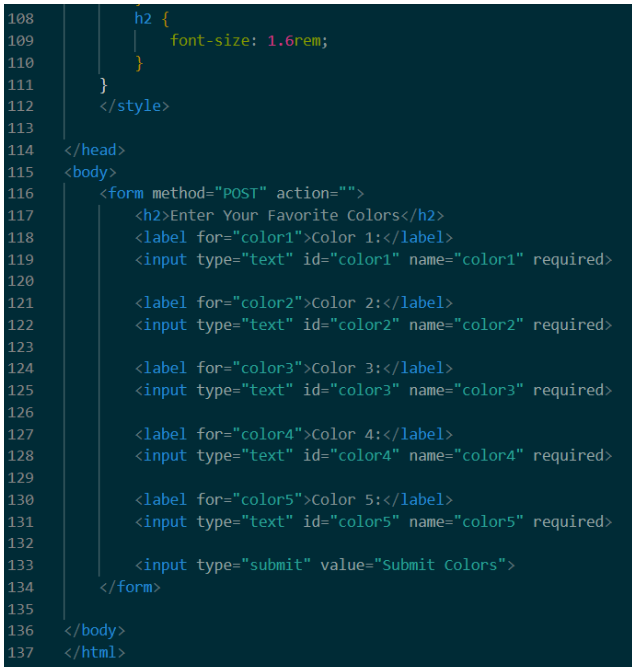
```php
<?php
session_start();
if($_SERVER['REQUEST_METHOD'] == 'POST'){
    $_SESSION['colors'] = [
        $_POST['color1'],
        $_POST['color2'],
        $_POST['color3'],
        $_POST['color4'],
        $_POST['color5']
    ];
    header("Location: displayColors.php");
    exit();
}
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Favorite Colors Session</title>
    <style>
    body {
        font-family: 'Comic Sans MS', cursive, sans-serif;
        background: linear-gradient(to right, #fdf0d5, #f9c5d1);
        padding: 40px 20px;
        text-align: center;
        color: #ff4081;
        min-height: 100vh;
        display: flex;
        justify-content: center;
        align-items: center;
    }

    form {
        background: #fff3f9;
        padding: 40px 50px;
```



```css
        border-radius: 25px;
        box-shadow: 0 12px 25px rgba(255, 105, 180, 0.3);
        max-width: 500px;
        width: 100%;
        border: 5px solid  #ff80ab;
        transition: transform 0.3s ease-in-out;
    }

    form:hover {
        transform: scale(1.02);
    }

    h2 {
        margin-bottom: 30px;
        font-weight: bold;
        font-size: 2rem;
        color: #d81b60;
        text-shadow: 1px 1px #ffc1e3;
    }

    label {
        display: block;
        margin: 16px 0 8px 0;
        font-weight: bold;
        font-size: 1.1rem;
        color: #ab47bc;
        text-align: left;
    }

    input[type="text"] {
        padding: 12px;
        width: 100%;
        border-radius: 12px;
        border: 2px dashed #ce93d8;
        font-size: 1rem;
```

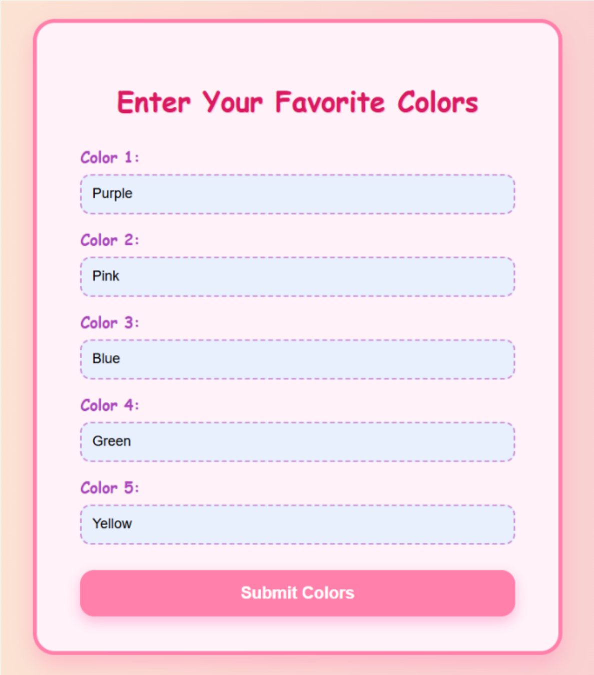


```
        background-color: #fff0f5;
        color: #6a1b9a;
        box-sizing: border-box;
        transition: all 0.3s;
    }

    input[type="text"]:focus {
        border-color: #f48fb1;
        background-color: #fce4ec;
        outline: none;
    }

    input[type="submit"] {
        background: #ff80ab;
        color: white;
        border: none;
        padding: 15px 0;
        font-weight: bold;
        border-radius: 16px;
        font-size: 1.2rem;
        margin-top: 30px;
        width: 100%;
        cursor: pointer;
        box-shadow: 0 6px 15px rgba(255, 105, 180, 0.4);
        transition: background 0.3s ease;
    }

    input[type="submit"]:hover {
        background: #f06292;
    }

    @media (max-width: 480px) {
        form {
            padding: 30px;
        }
```



```
        h2 {
            font-size: 1.6rem;
        }
    }
    </style>

</head>
<body>
    <form method="POST" action="">
        <h2>Enter Your Favorite Colors</h2>
        <label for="color1">Color 1:</label>
        <input type="text" id="color1" name="color1" required>

        <label for="color2">Color 2:</label>
        <input type="text" id="color2" name="color2" required>

        <label for="color3">Color 3:</label>
        <input type="text" id="color3" name="color3" required>

        <label for="color4">Color 4:</label>
        <input type="text" id="color4" name="color4" required>

        <label for="color5">Color 5:</label>
        <input type="text" id="color5" name="color5" required>

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

</body>
</html>
```

OUTPUT (colorSession.php)



Enter Your Favorite Colors

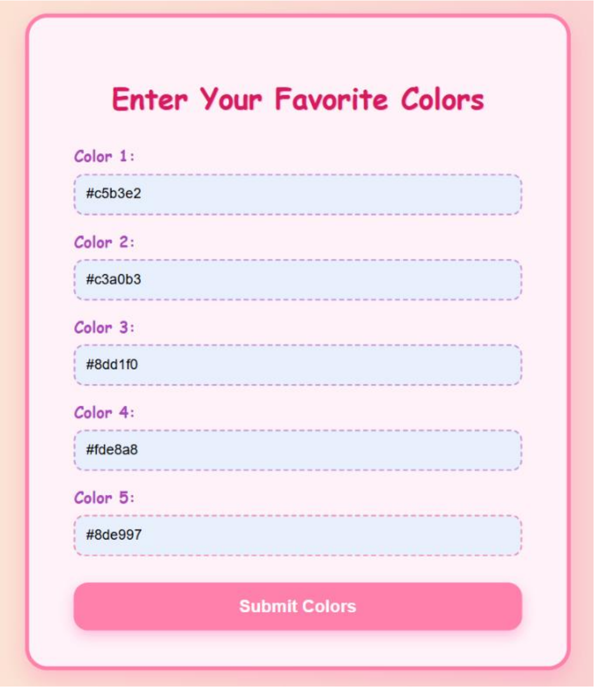
Color 1: Purple

Color 2: Pink

Color 3: Blue

Color 4: Green

Color 5: Yellow



Enter Your Favorite Colors

Color 1: #c5b3e2

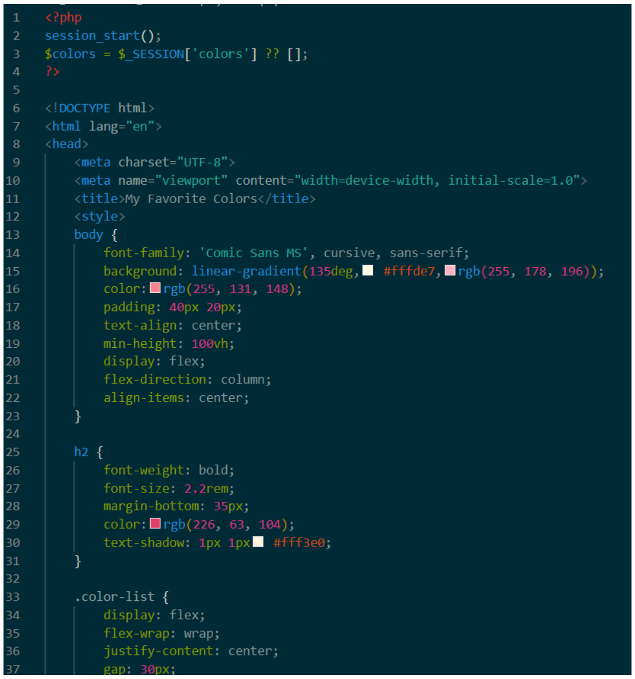
Color 2: #c3a0b3

Color 3: #8dd1f0

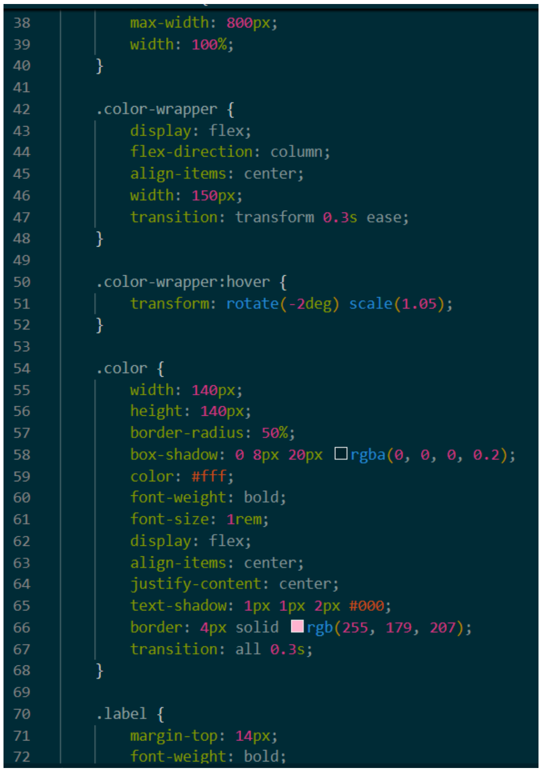
Color 4: #fde8a8

Color 5: #8de997

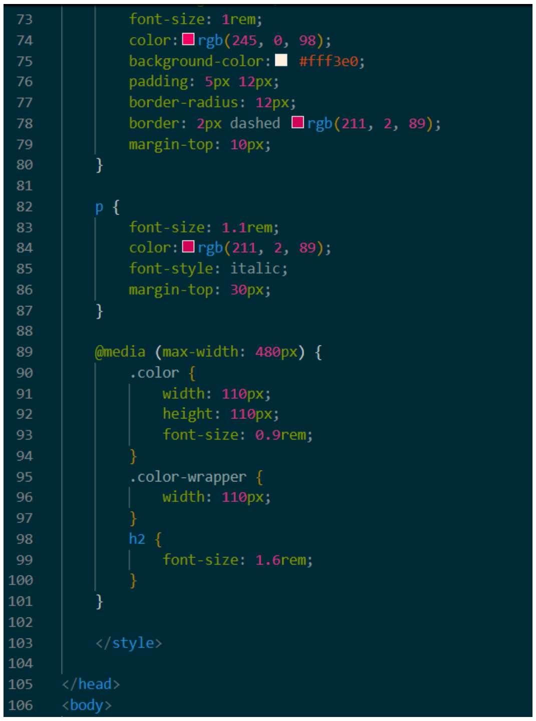
SOURCE CODE (displayColors.php)



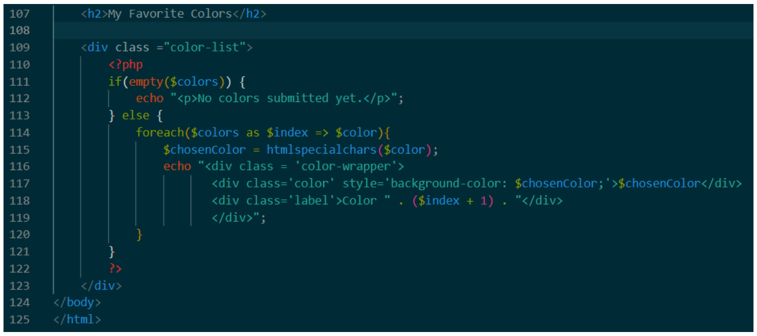
```php
<?php
session_start();
$colors = $_SESSION['colors'] ?? [];
?>

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>My Favorite Colors</title>
    <style>
    body {
        font-family: 'Comic Sans MS', cursive, sans-serif;
        background: linear-gradient(135deg, #fffde7, rgb(255, 178, 196));
        color: rgb(255, 131, 148);
        padding: 40px 20px;
        text-align: center;
        min-height: 100vh;
        display: flex;
        flex-direction: column;
        align-items: center;
    }

    h2 {
        font-weight: bold;
        font-size: 2.2rem;
        margin-bottom: 35px;
        color: rgb(226, 63, 104);
        text-shadow: 1px 1px #fff3e0;
    }

    .color-list {
        display: flex;
        flex-wrap: wrap;
        justify-content: center;
        gap: 30px;
```



```css
        max-width: 800px;
        width: 100%;
    }

    .color-wrapper {
        display: flex;
        flex-direction: column;
        align-items: center;
        width: 150px;
        transition: transform 0.3s ease;
    }

    .color-wrapper:hover {
        transform: rotate(-2deg) scale(1.05);
    }

    .color {
        width: 140px;
        height: 140px;
        border-radius: 50%;
        box-shadow: 0 8px 20px rgba(0, 0, 0, 0.2);
        color: #fff;
        font-weight: bold;
        font-size: 1rem;
        display: flex;
        align-items: center;
        justify-content: center;
        text-shadow: 1px 1px 2px #000;
        border: 4px solid rgb(255, 179, 207);
        transition: all 0.3s;
    }

    .label {
        margin-top: 14px;
        font-weight: bold;
```

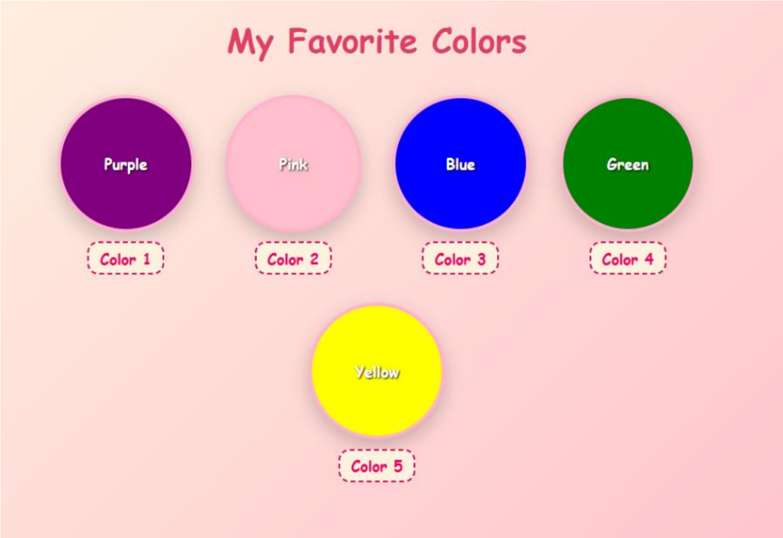


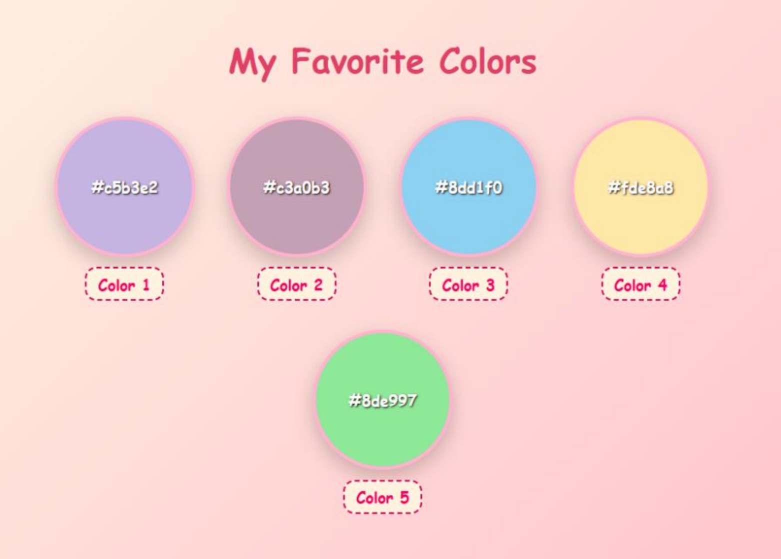
```css
        font-size: 1rem;
        color: rgb(245, 0, 98);
        background-color: #fff3e0;
        padding: 5px 12px;
        border-radius: 12px;
        border: 2px dashed rgb(211, 2, 89);
        margin-top: 10px;
    }

    p {
        font-size: 1.1rem;
        color: rgb(211, 2, 89);
        font-style: italic;
        margin-top: 30px;
    }

    @media (max-width: 480px) {
        .color {
            width: 110px;
            height: 110px;
            font-size: 0.9rem;
        }
        .color-wrapper {
            width: 110px;
        }
        h2 {
            font-size: 1.6rem;
        }
    }

    </style>

</head>
<body>
```



```php
    <h2>My Favorite Colors</h2>

    <div class ="color-list">
        <?php
        if(empty($colors)) {
            echo "<p>No colors submitted yet.</p>";
        } else {
            foreach($colors as $index => $color){
                $chosenColor = htmlspecialchars($color);
                echo "<div class = 'color-wrapper'>
                        <div class='color' style='background-color: $chosenColor;'>$chosenColor</div>
                        <div class='label'>Color " . ($index + 1) . "</div>
                        </div>";
            }
        }
        ?>
    </div>
</body>
</html>
```

OUTPUT (displayColors.php)





VII. QUESTION AND ANSWER

1. What is the usage of super global?
   -Superglobals are PHP variables that can be accessed from any scope throughout the whole code. It is used for collecting data such as inputs from forms, server information, and session data without passing the variables explicitly.

2. What are the differences between $_POST and $_GET?

-$_GET sends data through URL Parameters, that is why, the inputs from forms can be seen in the URL, hence it is also limited in size. While $_POST are passed through HTTP Post method which is more secure rather than the $_GET function.

3. What is the importance of cookies on a webpage or website?
   -Cookies store small pieces of data on the user's browser to remember preferences, login status, and track user behavior across sessions. They enable personalized user experiences and maintain state in stateless HTTP connections.

4. What is a session?
   -A session is a server-side storage mechanism that keeps user data across multiple page requests during a visit. It allows the website to remember user-specific information without exposing it to the client.

5. What is the importance of session?
   -Sessions provide a secure way to maintain user state and data, such as login credentials or shopping cart contents, throughout a browsing session. They enhance user experience by enabling continuity and personalization without relying solely on client-side storage.

VIII. REFERENCES

1. https://www.w3schools.com/css/
2. https://www.w3schools.com/html/
3. https://www.w3schools.com/php/php_variables.asp
4. https://www.w3schools.com/php/php_superglobals_globals.asp
5. https://www.w3schools.com/php/php_superglobals_post.asp
6. https://www.w3schools.com/php/php_superglobals_get.asp
7. https://www.w3schools.com/php/php_superglobals_request.asp
8. https://www.w3schools.com/php/php_cookies.asp
9. https://www.w3schools.com/php/php_sessions.asp

Note: The following rubrics/metrics will be used to grade students' output.

Program (100 pts.)	**(Excellent)**	**(Good)**	**(Fair)**	**(Poor)**
Program execution (20pts)	Program executes correctly with no syntax or runtime errors **(18-20pts)**	Program executes with less than 3 errors **(15-17pts)**	Program executes with more than 3 errors **(12-14pts)**	Program does not execute **(10-11pts)**
Correct output (20pts)	Program displays correct output with no errors **(18-20pts)**	Output has minor errors **(15-17pts)**	Output has multiple errors **(12-14pts)**	Output is incorrect **(10-11pts)**
Design of output (10pts)	Program displays more than expected **(10pts)**	Program displays minimally expected output **(8-9pts)**	Program does not display the required output **(6-7pts)**	Output is poorly designed **(5pts)**
Design of logic (20pts)	Program is logically well designed **(18-20pts)**	Program has slight logic errors that do no significantly affect the results **(15-17pts)**	Program has significant logic errors **(3-5pts)**	Program is incorrect **(10-11pts)**
Standards (20pts)	Program code is stylistically well designed **(18-20pts)**	Few inappropriate design choices (i.e. poor variable names, improper indentation) **(15-17pts)**	Several inappropriate design choices (i.e. poor variable names, improper indentation) **(12-14pts)**	Program is poorly written **(10-11pts)**
Delivery (10pts)	The program was delivered on time. **(10pts)**	The program was delivered a day after the deadline. **(8-9pts)**	The program was delivered two days after the deadline. **(6-7pts)**	The program was delivered more than two days after the deadline. **(5pts)**