Progressive Education Society's

**Modern College of Engineering, Pune**

**MCA Department**

**A.Y.2024-25**

**(410902) Web Technologies Lab**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Class: SY-MCA Shift / Div : A Roll Number : 52062

Name: Laxman Shinde Assignment No : 5 Batch: S3 Date : 19/09/2024

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Q.1.Create a PHP code, which will retrieve the data captured by the HTML form,

display the name of the form and also display the message Data Entered

Successfully on the HTML form page, after performing the following

validations

a. The book name form field is not left blank.

b. The author name form field is not left blank.

c. The publisher name form field is not left blank.

d. The synopsis form field is not left blank.

CODE :

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Book Form</title>

</head>

<body>

<h2>Book Form</h2>

<?php

// Define variables and set to empty values

$bookNameErr = $authorNameErr = $publisherNameErr = $synopsisErr = "";

$bookName = $authorName = $publisherName = $synopsis = "";

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

// Validate book name

if (empty($\_POST["bookName"])) {

$bookNameErr = "Book name is required";

} else {

$bookName = test\_input($\_POST["bookName"]);

}

// Validate author name

if (empty($\_POST["authorName"])) {

$authorNameErr = "Author name is required";

} else {

$authorName = test\_input($\_POST["authorName"]);

}

// Validate publisher name

if (empty($\_POST["publisherName"])) {

$publisherNameErr = "Publisher name is required";

} else {

$publisherName = test\_input($\_POST["publisherName"]);

}

// Validate synopsis

if (empty($\_POST["synopsis"])) {

$synopsisErr = "Synopsis is required";

} else {

$synopsis = test\_input($\_POST["synopsis"]);

}

}

function test\_input($data) {

$data = trim($data);

$data = stripslashes($data);

$data = htmlspecialchars($data);

return $data;

}

?>

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

Book Name: <input type="text" name="bookName">

<span class="error"><?php echo $bookNameErr;?></span>

<br><br>

Author Name: <input type="text" name="authorName">

<span class="error">\* <?php echo $authorNameErr;?></span>

<br><br>

Publisher Name: <input type="text" name="publisherName">

<span class="error">\* <?php echo $publisherNameErr;?></span>

<br><br>

Synopsis: <textarea name="synopsis" rows="5" cols="40"></textarea>

<span class="error">\* <?php echo $synopsisErr;?></span>

<br><br>

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

</form>

<?php

// Display data if all fields are filled

if ($bookName != "" && $authorName != "" && $publisherName != "" && $synopsis != "") {

echo "<h2>Data Entered Successfully</h2>";

echo "<p><strong>Book Name:</strong> $bookName</p>";

echo "<p><strong>Author Name:</strong> $authorName</p>";

echo "<p><strong>Publisher Name:</strong> $publisherName</p>";

echo "<p><strong>Synopsis:</strong> $synopsis</p>";

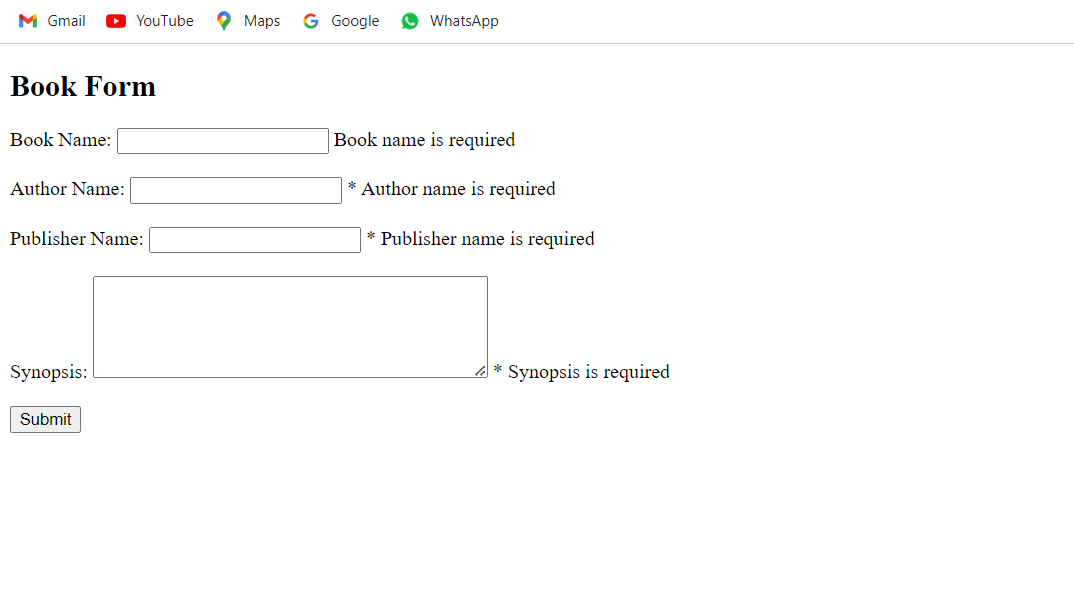
}

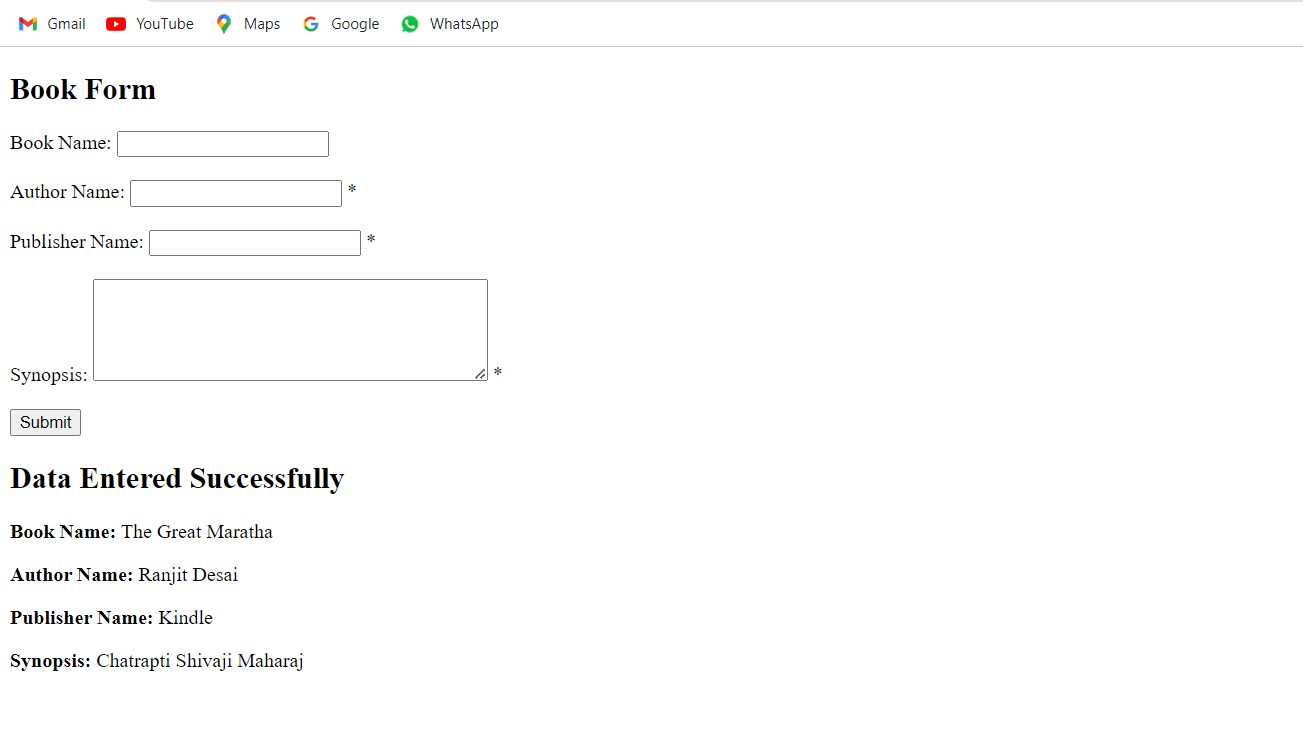
?>

</body>

</html>

OUTPUT :





**Q.2. Use Cookies to retain some information for 30 minutes, dependent on logged-in**

**users.**

**CODE :**

<?php

// Start the session

session\_start();

// Function to set a cookie with a 30-minute expiration time

function setCookieWithExpiration($name, $value, $expiration) {

setcookie($name, $value, time() + $expiration, "/");

}

// Function to check if the user is logged in

function isLoggedIn() {

return isset($\_SESSION['user\_id']); // Adjust the condition based on your actual login mechanism

}

// Check if the user is logged in

if (isLoggedIn()) {

// User is logged in, set a cookie with a 30-minute expiration time

setCookieWithExpiration('user\_info', 'some\_data', 1800); // 1800 seconds = 30 minutes

}

// Example of retrieving the cookie value

if (isset($\_COOKIE['user\_info'])) {

$userInfo = $\_COOKIE['user\_info'];

// Do something with $userInfo

} else {

// Cookie is not set, or it has expired

// Handle the case accordingly

}

?>

**Q.3. Create HTML page that contain textbox, submit / reset button. Write PHP**

**program to display this information and store into text file.**

**CODE :**

Index.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>Form Submission</title>

</head>

<body>

<form action="process\_form.php" method="post">

<label for="user\_input">Enter Information:</label>

<input type="text" id="user\_input" name="user\_input" required>

<br>

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

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

</form>

</body>

</html>

Process\_form.php

<?php

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

// Retrieve the user input from the form

$userInput = $\_POST["user\_input"];

// Validate and sanitize the input (you can add more robust validation)

$userInput = htmlspecialchars(trim($userInput));

// Open or create a text file for writing

$file = fopen("user\_data.txt", "a");

if ($file) {

// Write the user input to the file along with a timestamp

fwrite($file, "[" . date("Y-m-d H:i:s") . "] " . $userInput . PHP\_EOL);

// Close the file

fclose($file);

echo "Data has been successfully stored.";

} else {

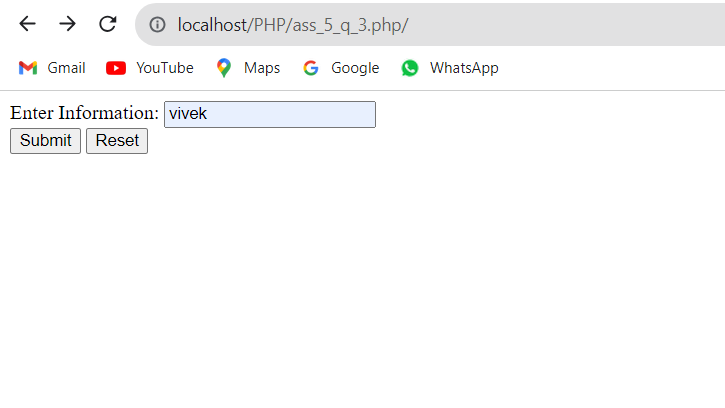
echo "Error: Unable to open the file for writing.";

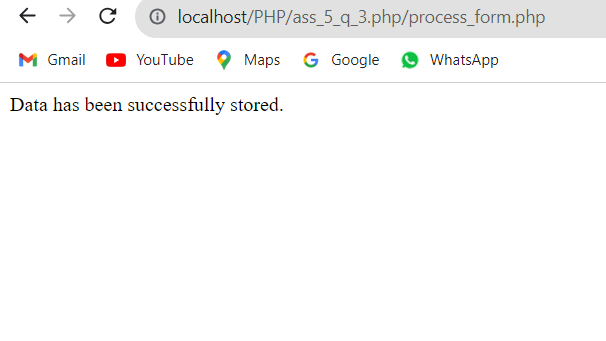
}

}

?>

OUTPUT :





**4. Create HTML page that contain textbox, submit / reset button. Write PHP**

**program to display this information and store into text file.**

**Code :**

**Index.html**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Simple Form</title>

</head>

<body>

    <h1>Enter Information</h1>

    <form action="submit.php" method="POST">

        <label for="info">Enter some text:</label><br>

        <input type="text" id="info" name="info" required><br><br>

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

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

    </form>

</body>

</html>

**Submit.php**

<?php

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

    $info = htmlspecialchars($\_POST['info']);

    echo "<h2>Information Submitted:</h2>";

    echo "<p>$info</p>";

    $file = 'data.txt';

    $current = file\_get\_contents($file);

    $current .= $info . "\n";

    file\_put\_contents($file, $current);

    echo "<p>Data has been saved to <strong>data.txt</strong>.</p>";

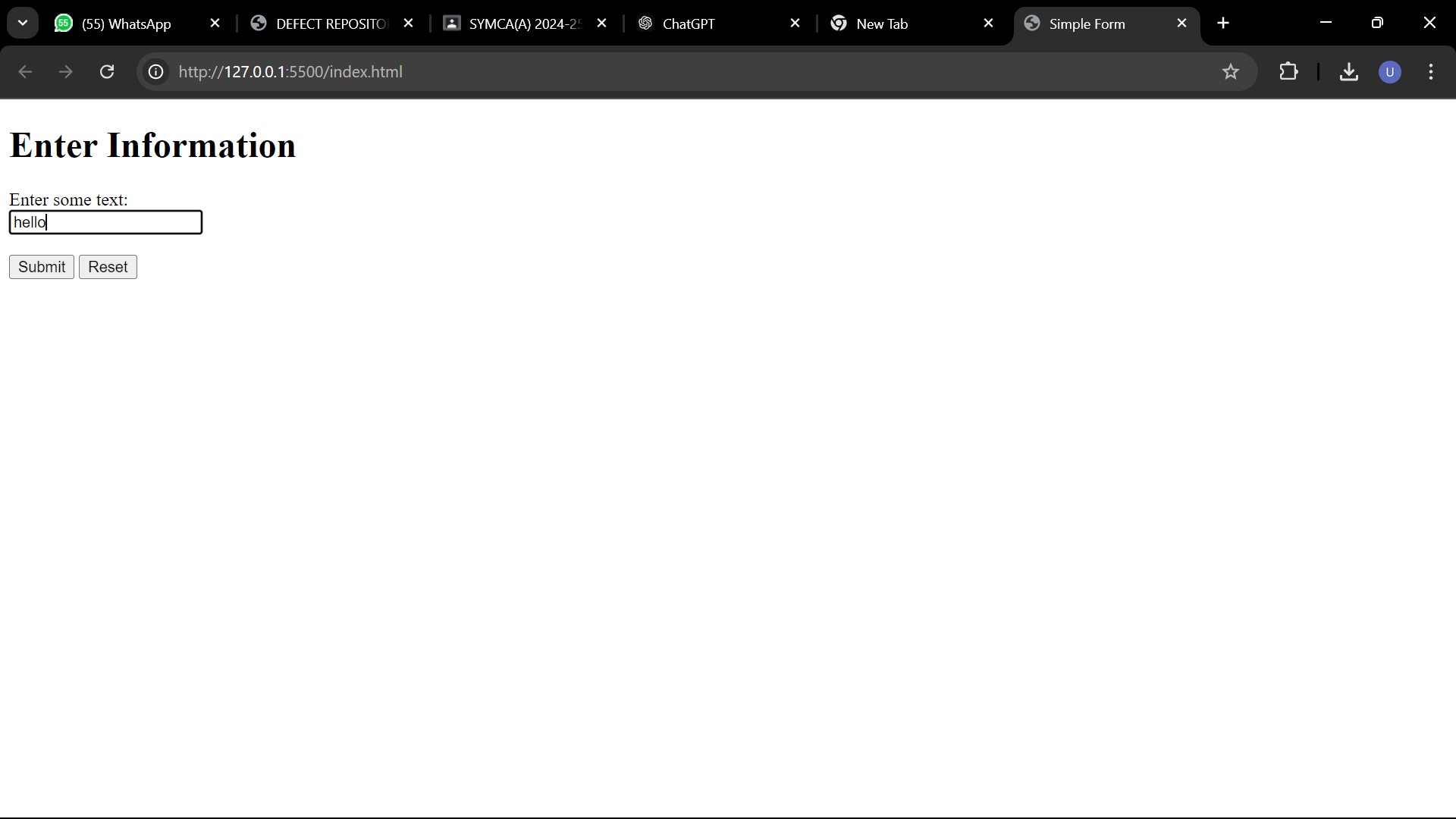
} else {

    echo "No data submitted.";

}

?>

**Output :**



**Q.5. Create an abstract class shape and also create Point, Circle, and Cylinder**

**hierarchy. Display the area of figures using polymorphism.**

**Code :**

<?php

abstract class Shape {

public abstract function area();

}

class Point extends Shape {

protected $x, $y;

public function \_\_construct($x, $y) {

$this->x = $x;

$this->y = $y;

}

public function area() {

return 0; // A point has no area

}

public function \_\_toString() {

return "Point(" . $this->x . ", " . $this->y . ")";

}

}

class Circle extends Point {

protected $radius;

public function \_\_construct($x, $y, $radius) {

parent::\_\_construct($x, $y);

$this->radius = $radius;

}

public function area() {

return pi() \* pow($this->radius, 2); // π \* r^2

}

public function \_\_toString() {

return "Circle(center = " . parent::\_\_toString() . ", radius = " . $this->radius . ")";

}

}

class Cylinder extends Circle {

private $height;

public function \_\_construct($x, $y, $radius, $height) {

parent::\_\_construct($x, $y, $radius);

$this->height = $height;

}

public function area() {

return 2 \* pi() \* $this->radius \* ($this->radius + $this->height); // 2πr(h + r)

}

public function \_\_toString() {

return "Cylinder(base = " . parent::\_\_toString() . ", height = " . $this->height . ")";

}

}

$shapes = [

new Point(1, 2),

new Circle(1, 2, 5),

new Cylinder(1, 2, 5, 10)

];

foreach ($shapes as $shape) {

echo "Shape: " . get\_class($shape) . "\n";

echo "Details: " . $shape . "\n";

echo "Area: " . number\_format($shape->area(), 2) . " square units\n";

echo str\_repeat("-", 40) . "\n";

}

