Experiment No. 6

Title: E-mail handling using PHP

Batch: B2 **Roll No.: 16010420117** **Experiment No.:7**

Aim: To perform E-mail handling using PHP

Resources needed: Windows OS, Web Browser, Editor, XAMPP Server

Pre-Lab/ Prior Concepts:
Students should have prior knowledge of HTML/CSS/Basic Programming.

Theory:

1. How to send Emails using PHP mail() Function

To use the PHP send mail feature, users hosting their PHP application or site on a local server and they will need to configure a **Sendmail** program by changing the **php.ini** file in their PHP installation folder. If you use a hosting server, **Sendmail** is usually already configured. However, you need to make sure that your hosting provider allows you to manually manage the **Sendmail service** option.

PHP is a widely used server side scripting language. It provides many built-in functions. PHP provides email support via a built-in mail() function. Using this function, you can easily send emails directly through your PHP script.

Email Syntax –

mail (to, subject, message, headers, parameters);

Here,

1. To: The first parameter which we have in email () function is 'to'. It is a mandatory parameter. You will have to give receiver's email id with this parameter.
2. Subject: It is also a mandatory field. It must be specified with the subject of the email. It can't contain newline characters.
3. Message: This field is also not optional. Message parameter has character limit of 70 characters so it should not exceed the limit. Moreover, the lines should be separated by (\n).
4. Headers: It is used to specify additional headers like BCC and CC. They must be separated by (/r/n). Form header is essential in sending emails. It is set with this parameter or by php.ini file.
5. Parameters: It can be used to set additional parameter like setting envelope sender address.

Example –

```
<?php
        $to = "yourname@yourwebsite.com";
        $headers = "From: $email_from \r\n";
        $headers = "Reply-To: $visitor_email \r\n";
        mail($to, $email_subject, $email_body, $headers);
 ?>
```

When we want to send the email to more than one recipient, then we just need to add these email addresses in the “$to” variable.

```
<?php
  $to = "name1@website-name.com,  name2@website-name.com,name3@website-name.com";
  mail($to, $email_subject, $email_body,$headers);
?>
```

We can use the CC (carbon copy) and BCC (Blind Carbon Copy) parameters as well. The CC and BCC emails are added in the ‘headers’ parameter.

Example –

```
<?php
    $to="name1@website-name.com, name2@website-name.com,name3@website-name.com";
    $headers = "From: $email_from \r\n";
    $headers = "Reply-To: $visitor_email \r\n";
    $headers= "Cc: someone@domain.com \r\n";
    $headers= "Bcc: someoneelse@domain.com \r\n";
    mail($to, $email_subject, $email_body, $headers);
 ?>
```

2. How to send Emails using PHP mail() Function

What is PHP Mailer?

PHPMailer is the classic email sending library for PHP. It supports several ways of sending email messages such as mail(), Sendmail, qmail, and direct dispatch to SMTP servers. In addition, it provides a list of advanced features:

- SMTP authentication
- secure/MIME encryption
- support of TLS and SSL protocols
- HTML content along with plain text
- multiple fs, string, and binary attachments
- embedded images support

PHPMailer is a code library and used to send emails safely and easily via PHP code from a web server. Sending emails directly via PHP code requires a high-level familiarity to SMTP standard protocol and related issues and vulnerabilities about Email injection for spamming. PHPMailer simplifies the process of sending emails and it is very easy to use.

Step-1: Installation of Library PHPMailer –

To install PHPMailer in PHP Project, run the below shown composer command –

```
composer require phpmailer / phpmailer
```

Step-2: Load the Composer’s Autoloader -

```
require ‘vendor/autoload.php’;
```

Step-3: Create a PHPMailer class object –

```
$mail = PHPmailer()
```

Step-4: Configure Server Setting –

```
$mail->SMTPDebug = 2;                          // Enable verbose debug output
$mail->isSMTP();                               // Set mailer to use SMTP
$mail->Host   = 'smtp.gfg.com;';               // Specify main SMTP server
$mail->SMTPAuth   = true;                      // Enable SMTP authentication
$mail->Username   = 'user@gfg.com';            // SMTP username
$mail->Password   = 'password';                // SMTP password
$mail->SMTPSecure = 'tls';                     // Enable TLS encryption, 'ssl' also accepted
$mail->Port   = 587;                           // TCP port to connect to
```

Activity:

- Implement Email functionality using SendEmail configuration or PHP Mailer library
- Functions should have simple message sending, use of CC feature and use of additionalheader parameters.

Output (Code with result Snapshot)

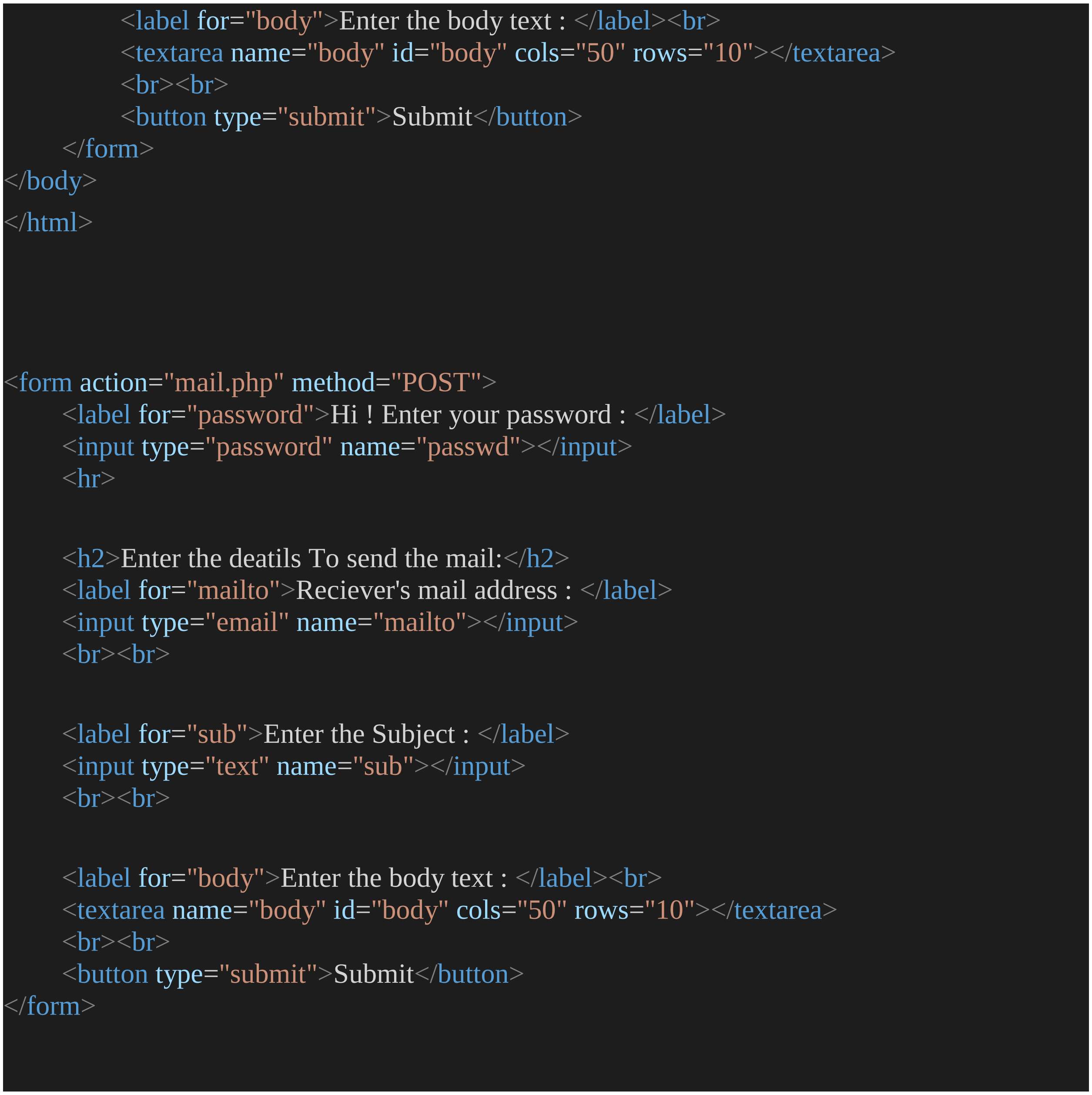
Command:

composer require phpmailer / phpmailer

Code:

Index.php

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>PHPMailer</title>
</head>
<body>
    <form action="mail.php" method="POST">
        <label for="password">Hi ! Enter your password : </label>
        <input type="password" name="passwd"></input>
        <hr>


        <h2>Enter the deatils To send the mail:</h2>
        <label for="mailto">Reciever's mail address : </label>
        <input type="email" name="mailto"></input>
        <br><br>


        <label for="sub">Enter the Subject : </label>
        <input type="text" name="sub"></input>
        <br><br>
```

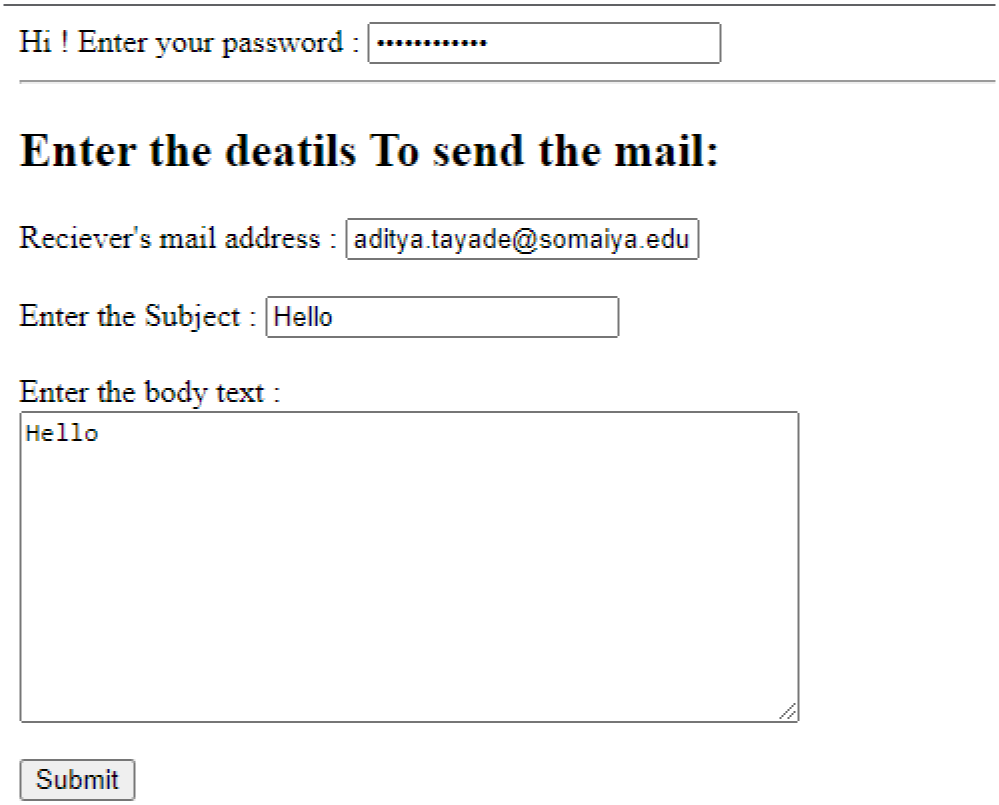


```
        <label for="body">Enter the body text : </label><br>
        <textarea name="body" id="body" cols="50" rows="10"></textarea>
        <br><br>
        <button type="submit">Submit</button>
    </form>
</body>
</html>




<form action="mail.php" method="POST">
    <label for="password">Hi ! Enter your password : </label>
    <input type="password" name="passwd"></input>
    <hr>


    <h2>Enter the deatils To send the mail:</h2>
    <label for="mailto">Reciever's mail address : </label>
    <input type="email" name="mailto"></input>
    <br><br>


    <label for="sub">Enter the Subject : </label>
    <input type="text" name="sub"></input>
    <br><br>


    <label for="body">Enter the body text : </label><br>
    <textarea name="body" id="body" cols="50" rows="10"></textarea>
    <br><br>
    <button type="submit">Submit</button>
</form>
```

Mail.php

```
<?php

use PHPMailer\PHPMailer\PHPMailer;use
PHPMailer\PHPMailer\Exception;

require 'vendor/autoload.php';

$mail = new PHPMailer(true);
$password = $_POST["passwd"];
$mailto = $_POST["mailto"];
$subject = $_POST["sub"];
$body = $_POST["body"];



try {
        $mail->isSMTP();
        $mail->Host       = 'ssl://smtp.gmail.com';
        $mail->SMTPAuth = true;
        $mail->Username = 'siddhant.mishra@somaiya.edu';
        $mail->Password = $password;
        $mail->SMTPSecure = 'tls';
        $mail->Port        = 465;


        $mail->setFrom('soham.bhoir@somaiya.edu ', $subject);
        $mail->addAddress($mailto);


        $mail->isHTML(true);
        $mail->Subject = $subject;
        $mail->Body = $body;
        $mail->send();
        echo "Mail has been sent successfully!";
} catch (Exception $e) {
        echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";
}
?>
```

Screenshot:



Hi ! Enter your password : ••••••••••••

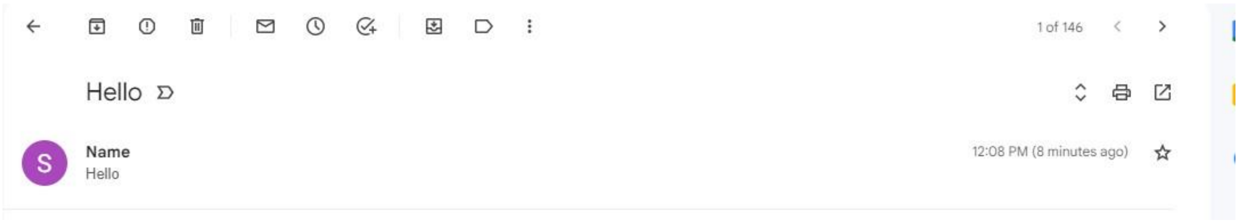
Enter the deatils To send the mail:

Reciever's mail address : aditya.tayade@somaiya.edu

Enter the Subject : Hello

Enter the body text :

Hello



Hello

Name

Hello

12:08 PM (8 minutes ago)

<u>Questions</u>: -

1. What are the advantages of using PHPMailer over mail() function to send mail in PHP?

Ans: First of all, PHPMailer provides an object-oriented interface, whereas mail() is not object oriented. PHP developers generally hate to create $headers strings while sending emails using the mail() function because they require a lot of escaping. PHPMailer makes this a breeze. Developers also need to write dirty code (escaping characters, encoding and formatting) to send attachments and HTML based emails when using the mail() function, whereas PHPMailer makes this painless.Also, the mail() function requires a local mail server to send out emails, which is not always trivial to set up. PHPMailer can use a non-local mail server (SMTP) if you have authentication.Further advantages include:

- It can print various kinds of error messages in more than 40 languages when it fails to send an email.
- It has integrated SMTP protocol support and authentication over SSL and TLS.
- It can send an alternative plain-text version of email for non-HTML email clients.

It has a very active developer community that keeps it secure and up to date.

2. How to use PHPMailer with Gmail?

Ans: The first step in using PHPMailer to send an HTML or plain text email is to mandate that PHPMailer utilize Composer. Open the command prompt in the root directory of your project, then enter: compose phpmailer / phpmailer

If the package is added successfully, the composer.json file should include the PHPMailer.

Go to Gmail, enable the less secure apps, disable two-factor authentication, and enter the aforementioned code. Your email will be sent successfully.

Outcomes: CO4: Demonstrate the use advanded features as REST API, email handling, localization, and internationalization in PHP.

Conclusion: (Conclusion to be based on the objectives and outcomesachieved)

Understood and successfully implemented the email functionality in PHP.

Grade: AA / AB / BB / BC / CC / CD/DD

Signature of faculty in-charge withdate

References:

Books:

1. Thomson PHP and MySQL Web Development Addison-Wesley Professional , 5th Edition 2016.
2. Peter MacIntyre, Kevin Tatroe Programming PHP O'Reilly Media, Inc, 4th Edition 2020
3. Frank M. Kromann Beginning PHP and MySQL: From Novice to Professional, Apress 1st Edition, 2018
4. https://www.w3schools.com/