

**Batch: A1 Roll No.: 1914018 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"](mailto: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

$to = "[name1@website-name.com,](mailto:name1@website-name.com) name2@website-name.com,name3@website-name.com"; mail($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,](mailto: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](mailto:someone@domain.com) \r\n";

$headers= "Bcc: [someoneelse@domain.com](mailto:someoneelse@domain.com) \r\n"; mail($to, $email\_subject, $email\_body, $headers);

?>

#### 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

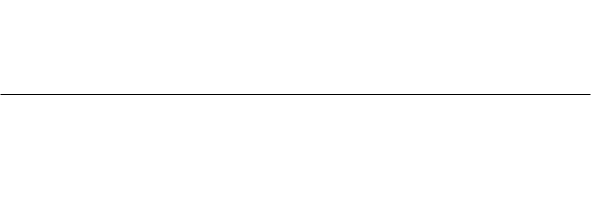
$mail->Username = 'user@gfg.com'; // SMTP username

$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 additional header parameters.

**Output (Code with result Snapshot)**

**CODE:**

**sendEmail.php**

<?php

    use PHPMailer\PHPMailer\PHPMailer;

    if (isset($\_POST['name']) && isset($\_POST['email'])) {

        $name = $\_POST['name'];

        $email = $\_POST['email'];

        $subject = $\_POST['subject'];

        $body = $\_POST['body'];

        require\_once "PHPMailer/PHPMailer.php";

        require\_once "PHPMailer/SMTP.php";

        require\_once "PHPMailer/Exception.php";

        $mail = new PHPMailer();

        $mail->isSMTP();

        $mail->Host = "smtp.gmail.com";

        $mail->SMTPAuth = true;

        $mail->Username = "urvik.gala@somaiya.edu"; //Your Email Address

        $mail->Password = 'urvikgala'; //Your Password

        $mail->Port = 465;

        $mail->SMTPSecure = "ssl";

        $mail->isHTML(true);

        $mail->setFrom($email, $name);

        $mail->addAddress("ravindradivekar@somaiya.edu"); //Your Email Address

        $mail->Subject = ("$email ($subject)");

        $mail->Body = $body;

        if ($mail->send()) {

            $status = "success";

            $response = "Email is sent!";

        } else {

            $status = "failed";

            $response = "Something is wrong: <br><br>" . $mail->ErrorInfo;

        }

        exit(json\_encode(array("status" => $status, "response" => $response)));

    }

?>

**Index.php**

<!DOCTYPE html>

<html>

<head>

    <title>Send an Email</title>

</head>

<body>

    <center>

        <h4 class="sent-notification"></h4>

        <form id="myForm">

            <h2>Send an Email</h2>

            <label>Name</label>

            <input id="name" type="text" placeholder="Enter Name">

            <br><br>

            <label>Email</label>

            <input id="email" type="text" placeholder="Enter Email">

            <br><br>

            <label>Subject</label>

            <input id="subject" type="text" placeholder=" Enter Subject">

            <br><br>

            <p>Message</p>

            <textarea id="body" rows="5" placeholder="Type Message"></textarea>

            <br><br>

            <button type="button" onclick="sendEmail()" value="Send An Email">Submit</button>

        </form>

    </center>

    <script src="http://code.jquery.com/jquery-3.3.1.min.js"></script>

    <script type="text/javascript">

        function sendEmail() {

            var name = $("#name");

            var email = $("#email");

            var subject = $("#subject");

            var body = $("#body");

            if (isNotEmpty(name) && isNotEmpty(email) && isNotEmpty(subject) && isNotEmpty(body)) {

                $.ajax({

                   url: 'sendEmail.php',

                   method: 'POST',

                   dataType: 'json',

                   data: {

                       name: name.val(),

                       email: email.val(),

                       subject: subject.val(),

                       body: body.val()

                   }, success: function (response) {

                        $('#myForm')[0].reset();

                        $('.sent-notification').text("Message Sent Successfully.");

                   }

                });

            }

        }

        function isNotEmpty(caller) {

            if (caller.val() == "") {

                caller.css('border', '1px solid red');

                return false;

            } else

                caller.css('border', '');

            return true;

        }

    </script>

</body>

</html>

# OUTPUT

# 

# 

**Questions: -**

* + 1. **What are the advantages of using PHPMailer over mail() function to send mail in PHP?**
       - Mail() function is not object oriented whereas PHPMailer possesses an interface which is object oriented.
       - In PHPmailer you don’t have to write dirty code. There is no need to create headers in the same way as in mail() function. PHPMailer actually makes life of a developer painless.

### PHPmailer supports sending plaintext email for non-HTML email clients whereas Mail() function does not support the same.

* + - * Mail() function usually needs local mail server for sending out emails whereas PHPMailer uses SMTP.

### ●

* + 1. **How to use PHPMailer with Gmail?**

1. Run the below command to install

**composer require phpmailer/phpmailer**

1. Include PHPMailer LIbrary/Class in code as below,

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

1. Include PHPMailer Library in PHP Script as below, ***require ‘path/to/PHPMailer/src/Exception.php’; require ‘path/to/PHPMailer/src/PHPMailer.php’; require ‘path/to/PHPMailer/src/SMTP.php’;***
2. SMTP

**isSMTP()** – Tell PHPMailer to use SMTP

**SMTPSecure** – Set the encryption system to use – ssl or tls **SMTPAuth**– set whether to use SMTP authentication (true/false) **Host** – Name of local or non-local mail server hostname

**Port-** Set the SMTP port number to connect to (587 for authenticated TLS)

**setFrom()** – Set who the mail is to be sent from

**addAddress()** – Set who the mail is to be sent to (Name of recipient is Optional)

## Outcomes:

Demonstrate the use of advanced features such as REST API, email handling, localization and Internationalization in PHP

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

Learned about email handling in PHP using Mail() Function and PHPMailer library, Implemented the same on project website

#### 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/>
5. [https://www.php.net/manual/en/function.mail.ph](https://www.php.net/manual/en/function.mail.php)