

Mass-Mail Dispatcher

PROJECT REPORT

Submitted By

Anitra Guin(ROLL NO:12620010002)

&

Monosij Chakraborty(ROLL NO:12620010022)

In Partial Fulfillment For 4th Semester Major Project

Of

MASTER OF COMPUTER APPLICATIONS

DEPARTMENT OF COMPUTER APPLICATIONS

HERITAGE INSTITUTE OF TECHNOLOGY, KOLKATA

**MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY,
WESTBENGAL**

JULY 2022

ACKNOWLEDGEMENT

I am highly indebted to my project guide Arvind Kumar (R&D Engineer of Exposys Data Labs) for guiding us and providing constant supervision and necessary information regarding the project & also for support in completing the Project.

I am thankful to the Department Of Computer Applications for providing us easy access to all the facilities and all the Professors For guiding and motivating us.

I am also thankful to Exposys Data Labs for giving us the opportunity To work on the project as an intern and providing us easy access to The project resources.



SIGNATURE

Anitra Guin

Roll No: 12620010002



SIGNATURE

Monosij Chakraborty

Roll No: 12620010022

Certificate



Declaration By Student

This is to declare that this report "Mass-Mail Dispatcher" has been Written by us. No part of the report is plagiarized from other sources. All information included from other sources have been duly acknowledged. I aver that if any part of the report is found to be plagiarized, I shall take full responsibility for it.

Anitra Guin.

SIGNATURE

Anitra Guin

Roll No: 12620010002

Monosij Chakraborty.

SIGNATURE

Monosij Chakraborty

Roll No: 12620010022

Abstract

Nowadays mailing takes very important part in our life. As a sense of organization we have to send mail every day to different persons. We always prefer mailing system for security of Data. So we create a unique concept of mailing system where we put all(50-60) the Recipients mail address in a single CSV file(sending Bulk mail) and we can send some document(Picture Or Word File) as well as to the selective recipients already added in CSV file. If some email have any wrong (Domain or any symbolic error) these email id are stored in another File and shows there as “invalid” and others are send to the recipients. So this system we create by using web development (PHP, PHP-Mailer & composer). This System will reduce time (we can send mail a list of person at a time through CSV file) as well as provide Security.

❖ Table Of Contents:

Introduction.....	7
Objectives of the System.....	7
Use Cases.....	8
Users of the System.....	8
Software Requirement Specifications.....	8
Functional Requirements.....	9
Non-Functional Requirements.....	9
Hardware & Software Requirements.....	10
Introduction To Mass-Mail Dispatcher.....	10
PHP Mailer.....	11
Composer.....	13
SMTP Works.....	10
Overview Of the System.....	15
Task 1.....	21
Task 2.....	23
Task 3.....	24
Code Screenshots.....	24-29
Tools Used.....	29-31

Introduction:

Millions Of mails are transferred every day, there may have some invalid mails. But in the process Of Mass-Mailing through CSV file it will reduce having invalid mails. The Actual goal of Mass-Mail is we can send huge bulk mails at a time without sending one by one to reduce the time consumption. This process is very efficient in our present situation.

Objective of the System:

The purpose of this project is to send huge bulk mail at a time. And also it can reduce having invalid mail. This process will save the time. We can perform some following task on this project:-

- I. Read the CSV file which contain huge no of mail ids.
- II. This system can send text, audio, video file using attachment facility.
- III. Check whether these CSV exists in system's file manager or not, if it doesn't exist then an error message will be generated.
- IV. Read the CSV file and detect the valid mail id and send mail to those mail id's successfully.

V. The system also detects the invalid mails and stores them into another CSV file for user's benefit.

Use Cases:

- **Send Mails Quickly:** This system can send mail fast.
- **Decrease In payload:** Removing the invalid mails and stores into another CSV file which saves lot of time.
- **Usability:** The PHP mailer library can be used by multiple users in order to send bulk mail with attachment.

Users of the system:

Any personnel (tester, developer, etc.) can send the bulk mails through the CSV file and if there is any defective mails in the CSV file then these defective mails will automatically store into another CSV file.

SOFTWARE REQUIREMENT SPECIFICATION

Software Requirement Specification (SRS) is comprehensive description of the environment for a system under development. The SRS fully describes what the system will do and how it is expected to perform.

Functional Requirements:

1. A CSV file is used to store bulk mails (including valid and invalid mails) that will send to recipients, which will be a main part of our project.
2. A PHP program which will read the CSV file and return a list of valid and invalid mails.
3. PHP mailer (Framework of PHP) is a method which is used in our system to send bulk mails easily and the whole process will be made very fast.
4. SMTP can login to our server by using the panel. It will navigate easily to all mail account. It will work with PHP mailer.
5. Composer makes an appropriate scripting command which can make the procedure faster.

Non-Functional Requirements:

The system should have the following Characteristics:

- Usable: The project should be efficient and user-friendly.
- Available: The project resource should always be available to the engineer so that he can put this system to any project and it will be compatible with all projects.
- Secure: This system should be protected against invalid mails.

Hardware and software Requirement Specifications:

Hardware Requirements:

- Processor: CPU 3.20 GHz.
- RAM: Minimum 4 GB.
- Storage: Minimum 500 GB.

Software Requirements:

- **Operating System:** Window 10 and higher.
- **Programming Language:** PHP, HTML, and CSS.
- **Documentation:** Microsoft office word (2007).
- **PPT (Presentation):** Microsoft power point (2007).

INTRODUCTION TO MASS-MAIL DISPATCHER

- **What is Mass-Mail Dispatcher?**

Mass-Mail Dispatcher is a system is use to send a bundle of bulk mails to number of recipients at a time. Now days it is very popular to send mails to huge recipients quickly. Now days every company can add this system with their website.

Features:-

- **Most Functionality:** Easy way to send bulk mail to huge recipients.
- **Send multi-media File:** As well as send bulk mails we can also easily send multi-media file (Audio, video, word file etc.) with attachment.
- **Fastest pace of innovation:** Now a days this process is made a fastest process for sending bulk mails to huge recipients at a time.

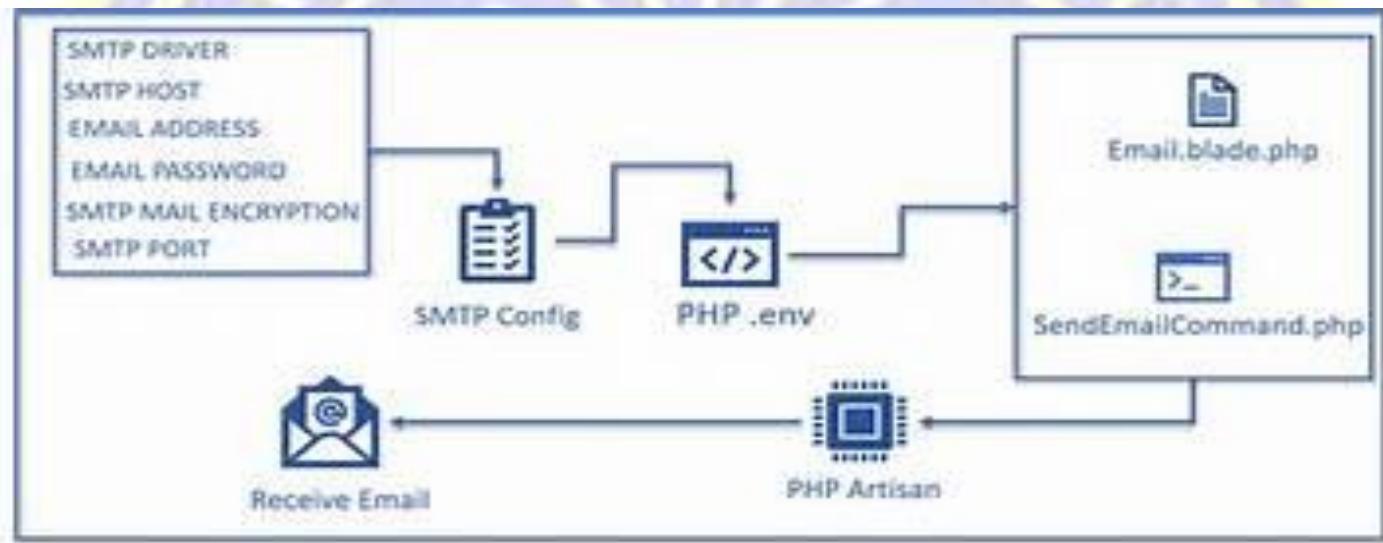
1. **PHP Mailer:** PHP mailer (Framework of PHP) is a method is use to send bulk mails easily and the whole process will make very fast. PHP mailer is a popular mail sending library of PHP. It supports mail sending via mail transfer protocol. To use PHP mail send features, users hosting the PHP application or a site on a local server will need to configure on a send mails program.



❖ Benefits of PHP Mailer:

- PHP Mailer makes it easy to change any parameter dynamically right in Our PHP script.
- PHP Mailer can be useful for if we send mail with attachment so it is work nicely.
- If it is not possible to send mails then PHP mailer will print error..
- It can send an alternative plain text version of mails client who don't have HTML.

❖ How It Works



❖ USE CASES:

- **SMTP authentication:** To test PHP mail functionality, we will use Mail trap, a fake SMTP server, so as not to flood our inboxes or even worse, the inboxes of our customers.
- **Secure/MIME encryption:** Using PHP mailer we can send mails securely.

- **Support of TLS and SSL protocols:** TLS simply uses a hash-based message authentication code for authentication, while SSL combines application data, PHP mailer supports this protocol to send message.
- **HTML content along with plain text:** PHPMailer provides an option to read an HTML message body from an external file, convert referenced images to embedded as well as convert HTML into a basic plain-text alternative body.
- **Multiple Documents with attachments:** By using PHP mailer we can send multiple documents with attachment.

2. **Composer:** Composer allows us to declare the libraries in our PHP projects, automatically load the code of the libraries, create and add necessary files to the PHP project and update libraries when have a new version. Composer makes an appropriate scripting command which can make the procedure faster.

❖ Benefits of Composer:

- **Tracks Every version of Package:** Composer tracks which versions of packages you have installed in a file called composer. Lock, which is intended

to be committed to version control, so that when the project is cloned in the future, simply running composer install will download and install all the project's dependencies.

- **Deals with PHP Dependencies:** composer deals with PHP dependencies on a per-project basis. This makes it easy to have several projects on one machine that depend on separate versions of one PHP package.
- **Provide Autoloader:** Composer provides an autoloader, making it extremely easy to get started with any package.
- **Automatically provide Update Version:** Composer allows us to easily update a project to the latest version that is allowed by your composer.json.

❖ How it is Works:

The Composer is designed to send simple, quick messages that do not contain a lot of images or complex layouts. Very much like a standard email that you send to a co-worker or friend. You can do basic text formatting, insert images, links, and emojis. The composer is perfect for those times when you don't need to send a more complex formatted message.

❖ Use Cases:

- **Provide Dependency Management:** Composer is a tool for dependency management in PHP. In simple words, Composer is used for managing libraries in your PHP project.
- **Declare Library:** Composer allows us to declare the libraries in your PHP projects, automatically load the code of the libraries, create and add

necessary files to the PHP projects, and update libraries when they have new versions.

- **Easily Search Package:** Packages is the primary repository for PHP Composer, and by default, the composer uses packagist.org to search for packages.

3. **SMTP (Simple Mail Transfer Protocol):** SMTP represents Simple Mail Transfer Protocol. SMTP is a set of interaction guidelines that allow the software to transmit electronic mail over the internet, referred to as Simple Mail Transfer Protocol. The main objective of SMTP is used to set up communication rules between servers. The servers have a way of recognizing themselves and announcing what kind of communication they are trying to perform. They also have a way of handling errors such as incorrect email address. SMTP can login to our server by using the panel. It will navigate easily to all mail account. It will work with PHP mailer.



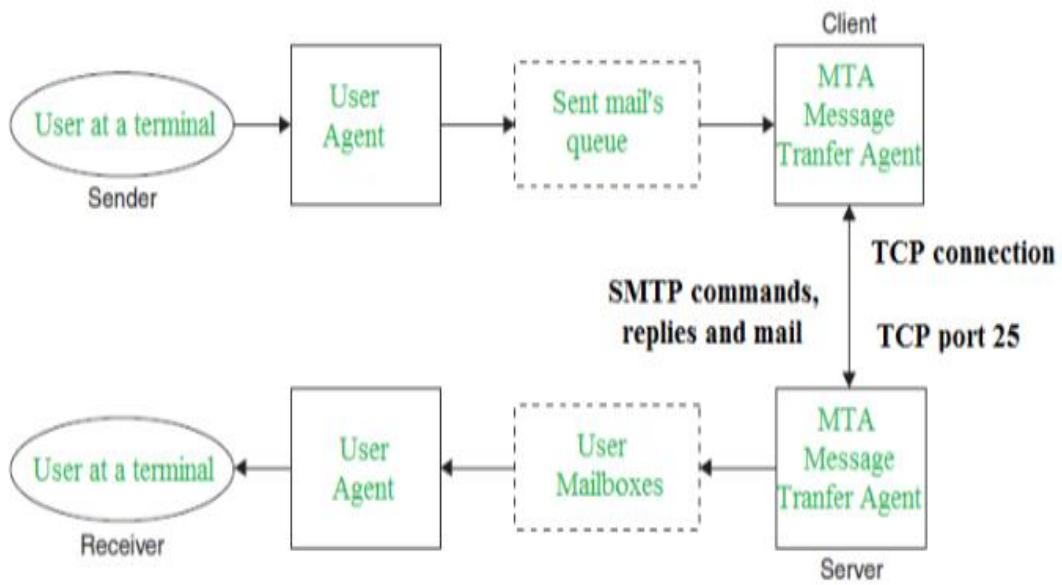
❖ Benefits Of SMTP:

- **Simplicity:** SMTP provides the simplest form of communicating through email messages between various computers in a particular network. Here are no hassles for the end-users as they only need to type their email and send it to the recipient's email address. The email will just go through a simple

process from the SMTP server to an exchange server for the recipient's computer making the electronic communication simple and not complicated.

- **Quick Mail Delivery:** Since SMTP is developed from a simple platform, email messages may be sent easily and quickly.
- **Reliability:** SMTP also offers reliability in terms of outgoing email messages. If there are instances where a particular message was not successfully sent, the SMTP server will always try to re-send the same email until the transmission becomes successful.
- **Option for dedicated servers:** With SMTP, companies and organizations may option to have a dedicated server to handle outgoing email messages. Although web mail services and ISPs may offer the same services, not many of them can actually provide exclusivity when it comes to mail servers. This could mean problems with email transmission. But with SMTP, servers are maintained locally and may be configured to exclusively handle very important and sensitive email messages.

❖How it Works:



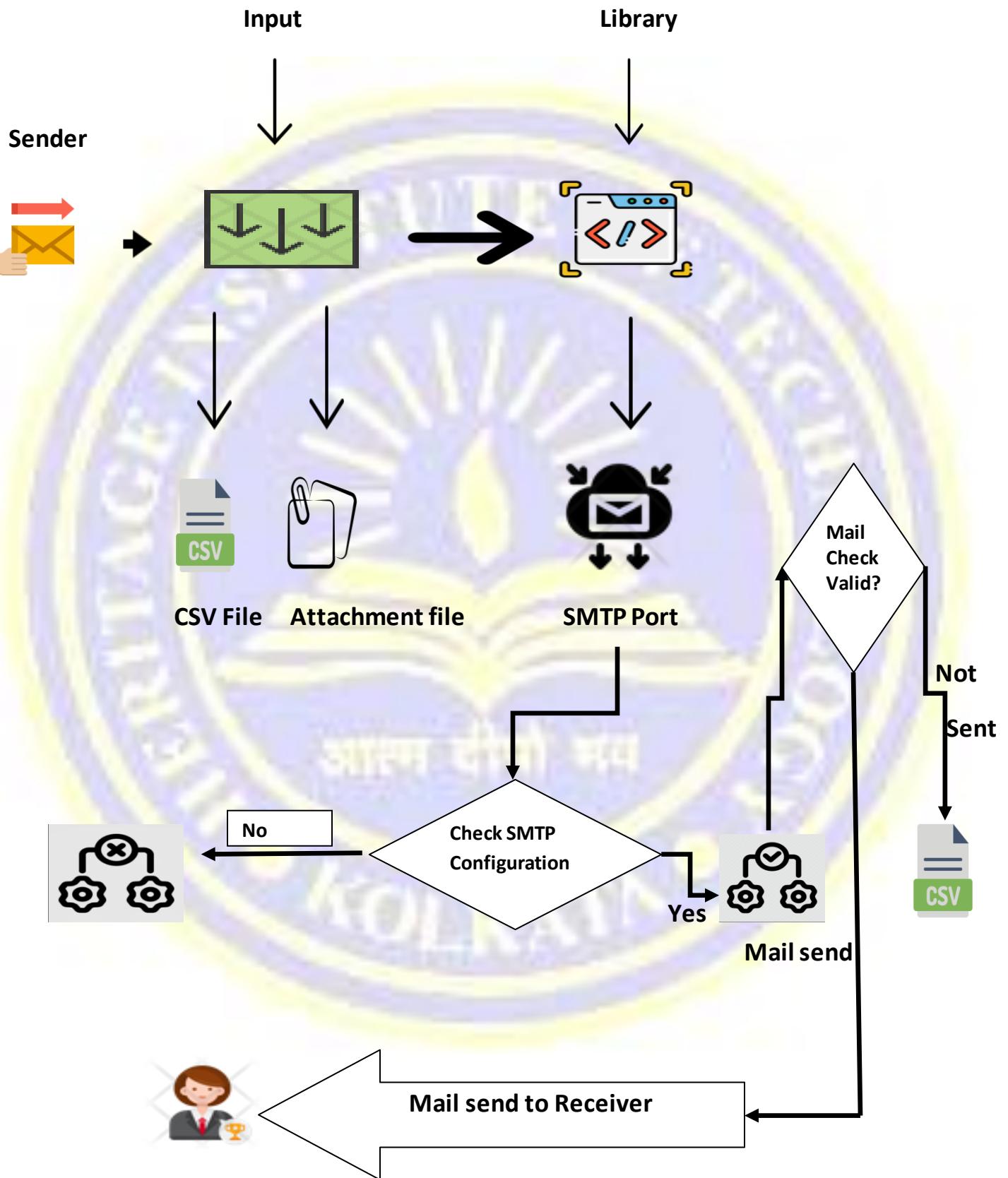
❖ Use Cases:

- **Communication Between Sender & Receiver:** The sender's user agent prepares the message and sends it to the MTA. The MTA's responsibility is to transfer the mail across the network to the receiver's MTA. To send mails, a system must have a client MTA, and to receive mails, a system must have a server MTA.
- **Sending Mail:** Mail is sent by a series of request and response messages between the client and the server. The message which is sent across consists of a header and a body.
- **Receiving Mail:** The user agent at the server-side checks the mailboxes at a particular time of intervals. If any information is received, it informs the user about the mail. When the user tries to read the mail it displays a list of emails with a short description of each mail in the mailbox. By selecting any of the mail users can view its contents on the terminal.

- **Mail Delivery:** The domain name and username of the receiver are the two parts of an email address. Let's assume, for instance, abcd@gmail.com, in which "abcd" is the username, and "gmail.com" is the domain. If the recipient's e-mail address does not coordinate with the sender's domain name, then the mail submission s agent (MSA) will transfer the mail (MTA).
- **Receipt & Processing:** After getting the incoming message, the exchange server passes it to the Mail Delivery Agent. It collects the e-mail and sets it for retrieval by the user.



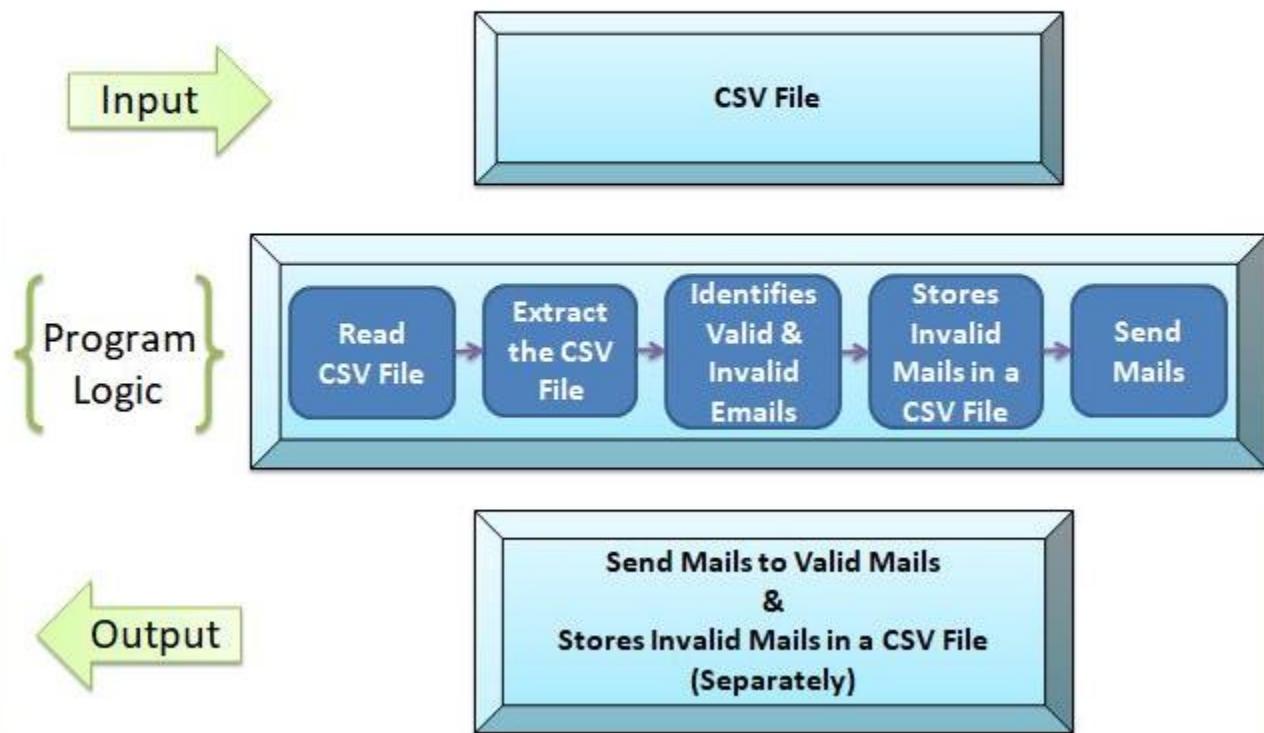
Overview Of the System



❖ **Concepts Of A System:** After doing Registration & Login successfully, every users get access to send huge number of mails using our website. Nowadays, it is very famous for many website which needs to send bulk mails at a single time in a day.

After getting information from the users we store those basic personal details into our database. So we created a “**Mass-Mailing System**” by adding this system into our website which helps them to send bulk mails easily at a time. So we created this system for Fast bulk mail send. The concepts is: - we write this code in **PHP** and **HTML** language and use some PHP library Like- **PHP mailer** is a Library of PHP, helps us to send bulk mail a huge no of Recipients at time. In our system we use PHP mailer for fast mail send. Also we use another Library of PHP i.e. **composer** it helps us to easily scripted our code in a machine. **SMTP** can send messages quickly in a server with port; by increasing the speed of port number it can send messages quickly. After all of these user's can put other recipients mail id into **CSV** file and our system check whether if there is any invalid mail id or not. If there is any invalid mail id it gets detected by our system and doesn't send message to those mail ids. These mail id stores in another CSV file. Our system will send message (also send doc or media file if required) to the valid mail ids only.

TASK 1: Identify Mails From CSV File



	A	B	C	D	E	F	G
1	anitraguin.11111@gmail.com						
2	anitra.guin.mca22@heritageit.edu.in						
3	monosij.chakraborty.mca22@heritageit.edu.in						
4	chakramanas1410@gmail.com						
5	nitaikrishnaguin@gmail.com						
6	anitaguin66@gmail.com						
7	chakrabortychitra20@gmail.com						
8	monodeadpool28@gmail.						
9	contact@odersin						
10	info.mp09@gmail.com						
11	anitraofficial@gmail.com						
12	jhipoffice@gmail.com						
13	mcscet@.com						
14	cite_rkl@rediffmail.com						
15	ksertcoe.com						
16	mc.marvel28@gmail.com						
17	dccomics98@gmail.com						
18	monosijc5@gmail.com						
19	itsmc98@gmail.com						
20	dataminingmca22@gmail.com						
21							
22							

Input .CSV
file

Note:

This CSV File Contains a mixture Of Valid & Invalid Mails. Our system takes this as input. Here we took an Example of 20 mails, but in real we can Deal with thousand of mails.

Task 2: Detect Invalid Mail

anitraguin.11111@gmail.com

anitra.guin.mca22@heritageit.edu.in

monosij.chakraborty.mca22@heritageit.edu.in

chakramanas1410@gmail.com

nitaikrishnaguin@gmail.com

anitaguin66@gmail.com

chakrabortychitra20@gmail.com

monodeadpool28@gmail. ---->>>This is an Invalid Email

contact@odersin ---->>>This is an Invalid Email

info.mp09@gmail.com ---->>>This is an Invalid Email

anitraofficial@gmail.com

jhipoffice@gmail.com

mcscet@.com ---->>>This is an Invalid Email

cite_rkl@rediffmail.com

ksertcoe.com ---->>>This is an Invalid Email

mc.marve128@gmail.com

dccomics98@gmail.com

monosijc5@gmail.com

itsmc98@gmail.com

dataminingmca22@gmail.com ---->>>This is an Invalid Email

Message has been sent

Task 3: Invalid Mails store in CSV file, Invalid.CSV

	A	B	C	D	E	F	G	H
1	monodeadpool28@gmail.							
2	contact@odersin							
3	info.mp09@gmail.com							
4	mcsct@.com							
5	ksertcoe.com							
6	dataminingmca22@gmail.com							
7								
8								
9								
10								
11								
12								
13								
14								
15								

Screen Shot Of Codes-

Server Settings

```
24 try {
25     //Server settings
26     $mail->SMTPDebug = 0;                                //Enable verbose debug output
27     $mail->isSMTP();                                     //Send using SMTP
28     $mail->Host      = 'smtp.gmail.com';                //Set the SMTP server to send through
29     $mail->SMTPAuth  = true;                             //Enable SMTP authentication
30     $mail->Username   = $sender;                          //SMTP username
31     $mail->Password   = $sender_password;               //SMTP password
32     $mail->SMTPSecure = PHPMailer::ENCRYPTION_SMTPS; //Enable implicit TLS encryption
33     $mail->Port       = 587;                            //TCP port to connect to; use 587
34     $mail->Port       = 465;                            //SSL port
```

Sending Mails Using PHP Mailer Instance and Separating Valid And Invalid mails

```
51 //Sending mails using PHPMailer Instance and seperating valid and invalid mails
52 $csv = $fileName;
53 $fh = fopen($csv, 'r');
54 $invalid = fopen("invalid.csv", 'a');
55     while(list($email) = fgetcsv($fh, 1024, ',')){
56         $sntz_email = filter_var($email, FILTER_SANITIZE_EMAIL);
57         if(filter_var($sntz_email, FILTER_VALIDATE_EMAIL)){
58             printf("<p> %s </p>", $sntz_email);
59             $mail->addAddress($sntz_email);
60
61         }else{
62             print($sntz_email." ---->>>This is an Invalid Email <br>");
63
64             $form_data = array(
65                 'email' => $sntz_email
66             );
67             fputcsv($invalid, $form_data);
68
69         }
70     }
71 }
```

Media File Attachments

```
73 //Attachments
74 for ($i=0; $i < count($attachments); $i++) {
75     $file_tmp = $_FILES["attachments"]["tmp_name"][$i];
76     $file_name = $_FILES["attachments"]["name"][$i];
77     move_uploaded_file($file_tmp, "attachments/" . $file_name);
78     $mail->addAttachment("attachments/" . $file_name);
79 }
80 }
```

Adding Content Mail

```
84     //Content
85     $mail->isHTML(true);           //Set email format to HTML
86     $mail->Subject = $subject;
87     $mail->Body      = $body;
88
89
90     $mail->send();
91     echo 'Message has been sent';
92 } catch (Exception $e) {
93     echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";
94 }
95
96 }
97 ?>
```

Index Form For Mail Sending

```
<h1 class="text-center fw-bold text-dark mb-3">Welcome <?php echo $_SESSION['username']; ?></h1>
<h2 class="text-center fw-bold text-dark mb-3">Email Sender</h2>
<form action="send.php" method="post" enctype="multipart/form-data">
    <div class="row">
        <div class="col-6 mb-3">
            <label for="sender_name" class="form-label">Sender Name</label>
            <input type="text" class="form-control" id="sender_name" name="sender_name" value="<?php echo $_SESSION['username']; ?>" placeholder="John Doe" readonly>
        </div>
        <div class="col-6 mb-3">
            <label for="sender" class="form-label">Sender Email</label>
            <input type="email" class="form-control" id="sender" name="sender" placeholder="name@example.com" required>
        </div>
        <div class="col-6 mb-3">
            <label for="sender_password" class="form-label">Sender Password</label>
            <input type="password" class="form-control" id="sender_password" name="sender_password" placeholder="Enter your email password" required>
        </div>
        <div class="col-6 mb-3">
            <label for="subject" class="form-label">Subject</label>
            <input type="text" class="form-control" id="subject" name="subject" placeholder="How are you?" required>
        </div>
        <div class="col-6">
            <label for="attachments" class="form-label">Attachments (Multiple)</label>
            <input type="file" class="form-control" multiple id="attachments" name="attachments[]" placeholder="name@example.com">
        </div>
    </div>
    <div class="mb-3">
        <label for="recipient" class="form-label">Recipient Emails</label>
        <input type="file" class="form-control" multiple id="recipient[]" name="recipient[]">
    </div>
    <div class="mb-3">
        <label for="body" class="form-label">Body</label>
        <textarea class="form-control" id="body" name="body" placeholder="Hi, How are you?" rows="5" required></textarea>
    </div>
    <div>
        <button class="btn btn-primary me-2" name="send" type="submit">Send Email</button>
        <button class="btn btn-danger" type="reset">Reset Form</button>
        <input type="button" onclick="location.href='user_login.php'" value="Log-out" style="position: relative; left: 62%">
    </div>
</form>
```

Activate Wind
Go to Settings to a

Normal Database Connection

```
1 <?php  
2     $db = mysqli_connect('localhost', 'root', '') or  
3         die ('Unable to connect. Check your connection parameters.' );  
4     mysqli_select_db($db, 'internship-project2' ) or die(mysqli_error($db));  
5  
6 ?>
```

Database Connection By Using PDO

```
1 <?php  
2  
3 $conn = new PDO("mysql:host=localhost; dbname=internship-project2;","root","");
4     $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
5 ?>
6
```

Login-Form

```
52             <form action="" method="post">
53                 <fieldset>
54                     <legend class="text-center fw-bold text-dark mb-3">Log-in</legend>
55                     <label for="uname">User-name</label>
56                     <input type = "text" name = "name" id="uname" placeholder="Enter your Email here"><br><br>
57
58                     <label for="upasswrd">Password:</label>
59                     <input type="password" name="password" id="upasswrd" placeholder="Enter your Password here"><br><br>
60                     <input type="submit" name="login" value="Login">
61
62                     <br><br>
63                     <p class="text-center text-dark mb-2">Create Account...</p>
64                     <center><input type="button" onclick="location.href='sign-up.php'" value="Sign-up"></center>
65                 </fieldset>
66             </form>
```

User Authentication From Database

```
5   if($_SERVER['REQUEST_METHOD']=='POST'){
6     if(!empty($_POST['login'])){
7       $name = $_POST['name'];
8       $password = md5($_POST['password']);
9
10    $stmt = "SELECT * FROM user_info WHERE
11          name = '$name' AND password = '$password'";
12
13    $result = mysqli_query($db,$stmt);
14    $count = mysqli_num_rows($result);
15    if($count > 0){
16      echo "<script>alert('Login Successful')</script>"; ?>
17
18      <META HTTP-EQUIV="Refresh" CONTENT="0; URL=http://localhost/internship-project3/index.html">
19      <?php
20      //echo $result['name'];
21      $_SESSION['username'] = $name;
22      header('location:index.php');
23      //header("Location:admin_show.php");
24    }else{
25      echo "<script>alert('Login not Successful')</script>"; ?>
26
27      <META HTTP-EQUIV="Refresh" CONTENT="0; URL=http://localhost/internship_project3/user_login.php">
28      <?php
29    }
30  }
```

Registration Form

```
61 <legend class="text-center fw-bold text-dark mb-3">Personel Details</legend>
62 <div class="col-6 mb-4">
63   <label for="sname">Name</label>
64   <input type="text" name="name" id="sname" placeholder="Enter your Name here"><br>
65 </div>
66 <div class="col-6 mb-4">
67   <label for="semail">Email</label>
68   <input type="text" name="email" id="semail" placeholder="Enter your Email here"><br>
69 </div>
70 <div class="col-6 mb-4">
71   <label for="spnumber">Contact number</label>
72   <input type="number" name="ph_number" id="spnumber" placeholder="Enter your phone number here"><br>
73 </div>
74 <div class="col-6 mb-4">
75   <p>Choose your Gender:- </p>
76   <input type="radio" name="gender" id="male" value="Male">
77   <label for="male">Male</label>
78   <input type="radio" name="gender" id="female" value="Female">
79   <label for="female">Female</label><br>
80 </div>
81 <div class="col-6 mb-4">
82   <label for="passwd">Create Password:</label>
83   <input type="password" name="password" id="passwd" placeholder="Enter your password here"><br>
84 </div>
```

Saving User informations From Registration Form to Database

```
6     if($_SERVER['REQUEST_METHOD']=='POST'){
7
8         $sname = $_POST['name'];
9
10        $semail = $_POST['email'];
11        $snumber = $_POST['ph_number'];
12        $sgender = $_POST['gender'];
13        $spassword = $_POST['password'];
14        $scpassword = $_POST['c_password'];
15        $sdob = $_POST['DOB'];
16
17
18        $stmt = $conn->prepare("INSERT INTO user_info (name,email_id,contact,gender,password,c_password,
19                                DOB) VALUES
20                                (:a,:b ,:c ,:d , :e, :f, :g)");
21
22        $stmt->bindValue(':a',$sname);
23        $stmt->bindValue(':b',$semail);
24        $stmt->bindValue(':c',$snumber);
25        $stmt->bindValue(':d',$sgender);
26        $stmt->bindValue(':e',md5($spassword));
27        $stmt->bindValue(':f',md5($scpassword));
28        $stmt->bindValue(':g',$sdob);
29
30
31        $stmt->execute();
32
33        echo "<script>alert('Record Created')</script>";
```

Tools Used:



VS-Code (Visual Studio Code) For Web-Developers-

Visual Studio Code is “a free-editor that helps the programmer write code helps in debugging and corrects the code using the intelli-sense method”. In normal

terms, it facilitates users to write the code in an easy manner. Many people say that it is half of an IDE and an editor, but the decision is up to the coders. Any program/software that we see or use works on the code that runs in the background. Traditionally coding was used to do in the traditional editors or even in the basic editors like notepad! These editors used to provide basic support to the coders.



PHP (Hypertext Preprocessor)- PHP is a programming language that allows web developers to write scripts that generate HTML pages dynamically. PHP was originally developed as a server-side scripting language, meaning that it runs on the server before being sent to the browser. However, with the advent of PHP 5, PHP has become more popular as a client-side scripting language. Client-side scripting languages run on the browser, so the code is executed directly to the browser.



HTML (Hypertext Markup Language)- HTML is an acronym which stands for Hyper Text Markup Language which is used for creating web pages and web applications. HTML is a markup language which is used for creating attractive web pages with the help of styling, and which looks in a nice format on a web browser. An HTML document is made of many HTML tags and each HTML tag contains different content.



CSS(Cascading Style Sheet): Cascading Style Sheets, fondly referred to as CSS, is a simple design language intended to simplify the process of making web pages presentable. CSS handles the look and feel part of a web page. Using CSS, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, and variations in display for different devices and screen sizes as well as a variety of other effects.

References

- PHP Function Basics:

https://r.search.yahoo.com/_ylt=AwrxgzO4sLxiuyUAMQfnHgx.;_ylu=Y29sbwMEcG9zAzIEdnRpZAMEc2VjA3Ny/RV=2/RE=1656561977/RO=10/RU=http%3a%2f%2fwww.w3schools.com%2fphp%2fphp_functions.asp/RK=2/RS=WXGcFmt1ih627D9kOMCe9Ni3ITw-

- PHP mailer tutorial:

https://r.search.yahoo.com/_ylt=AwrxhWp1sbxiLzUA4RPnHgx.;_ylu=Y29sbwMEcG9zAzMEDnRpZAMEc2VjA3Ny/RV=2/RE=1656562166/RO=10/RU=https%3a%2f%2fmailtrap.io%2fblog%2fphpmailer%2f/RK=2/RS=AzJjl9M5WGeoIWL5acd7ZsWQ4Q-

- PHP mailer Send Mails Basics:

https://r.search.yahoo.com/_ylt=AwrxhWp1sbxiLzUA2hPnHgx.;_ylu=Y29sbwMEcG9zAzIEdnRpZAMEc2VjA3Ny/RV=2/RE=1656562166/RO=10/RU=http

[s%3a%2f%2fwww.hostinger.com%2ftutorials%2fsend-emails-using-php-mail/RK=2/RS=uRsbIgRTTxjT7ZPKgt8IQd2sSjA-](https://www.hostinger.com/tutorials/send-emails-using-php-mail/RK=2/RS=uRsbIgRTTxjT7ZPKgt8IQd2sSjA-)

- Basic Of Composer:

https://r.search.yahoo.com/_ylt=AwrxguopsrxilnsAlhTnHgx.;_ylu=Y29sbwMEcG9zAzEEdnRpZAMEc2VjA3Ny/RV=2/RE=1656562346/RO=10/RU=https%3a%2f%2fw3schools.in%2fphp%2fphp-composer/RK=2/RS=uUDR4imwfNNVh6xJg1MENj8h5ag-

- HTML Function mail dispatcher:

https://r.search.yahoo.com/_ylt=Awrxguqosrxil704AMBpnhgx.;_ylu=Y29sbwMEcG9zAzIEdnRpZAMEc2VjA3Ny/RV=2/RE=1656562473/RO=10/RU=http%3a%2f%2fblog.hubspot.com%2fmarketing%2fhtml-form-email/RK=2/RS=dcghtZzcVLo7kJ5hWHRUpiMtNA-

- Use of SMTP Port:

https://r.search.yahoo.com/_ylt=AwrwBpVls7xiNx4AexDnHgx.;_ylu=Y29sbwMEcG9zAzUEdnRpZAMEc2VjA3Ny/RV=2/RE=1656562633/RO=10/RU=h%3a%2f%2fkinsta.com%2fblog%2fsmtp-port%2f/RK=2/RS=4G5kidhK8hHrDGvm19dtLqleoZA-

- Use of CSS:

https://r.search.yahoo.com/_ylt=AwrxhWqJubxiJRgAUwfnHgx.;_ylu=Y29sbwMEcG9zAzIEdnRpZAMEc2VjA3Ny/RV=2/RE=1656564233/RO=10/RU=http%3a%2f%2fwww.w3schools.com%2fw3css%2fw3css_templates.asp/RK=2/RS=tiMayTmM9Ai4SRh6NKZDKvulZUA-



