# PROFESSIONAL TRAINING REPORT

**at**

**Sathyabama Institute of Science and Technology**

**(Deemed to be University)**

Submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering Degree in Computer Science and Engineering

By

**PINNINTI UDAY VENKAT SARAN**

**REG. NO. 39110774**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**SCHOOL OF COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,**

**CHENNAI – 600119, TAMILNADU**

**APRIL 2022**

|  |  |  |
| --- | --- | --- |
|  | **SATHYABAMA**  **INSTITUTE OF SCIENCE AND TECHNOLOGY** (DEEMED TO BE UNIVERSITY) **Accredited with Grade “A” by NAAC**  (Established under Section 3 of UGC Act, 1956)  JEPPIAAR NAGAR, RAJIV GANDHI SALAI  CHENNAI– 600119  [**www.sathyabama.ac.in**](http://www.sathyabama.ac.in) |  |



# DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

**BONAFIDE CERTIFICATE**

This is to certify that this Project Report is the bonafide work of **PINNINTI UDAY VENKAT SARAN (Reg. No: 39110774)** who carried out the project entitled “**PASSWORD VERIFICATION USING PHP AND XAMPP**” under my supervision from March 2022 to April 2022.

## **Internal Guide**

**Dr. D. Usha Nandhini**

**Head of the Department**

**Dr.L. Lakshmanan, M.E., Ph.D.,**

**Dr.S. Vigneshwari,M.E.,Ph.D.,**



## Submitted for Viva voce Examination held on

**Internal Examiner External Examiner**

**DECLARATION**

I, UDAY VENKAT SARAN .P (Reg. No. 39110774) hereby declare that the Project report entitled **"PASSWORD VERIFICATION USING PHP AND XAMPP"** done by me under the guidance of **Dr. D. USHA NANDHINI** is submitted in partial fulfillment of the requirements for the award of Bachelor of degree in Information Technology.

### DATE:

**PLACE: SIGNATURE OF THE CANDIDATE**

### ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind

encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.SASIKALA, M.E., Ph.D., Dr. S. Vigneshwari , M.E., Ph.D. and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of **Computer Science** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. D. USHA NANDHINI,** for her valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department OF COMPUTER SCIENCE AND ENGINEERING** who were helpful in many ways for the completion of the project.

**TRAINING CERTIFICATE**

### TABLE OF CONTENTS

**CHAPTER TOPIC Page no**

[**ABSTRACT 9**](#_heading=h.gjdgxs)

[**LIST OF FIGURES**](#_heading=h.30j0zll)  **10**

**LIST OF ABBREVIATIONS 12**

**1. INTRODUCTION**

[WHAT IS PASSWORD VERIFICATION](#_heading=h.tyjcwt) **13**

[SYSTEM?](#_heading=h.tyjcwt)

[ROLE OF HTML, CSS, PHP & MYSQL **14**](#_heading=h.3dy6vkm)

### 

### REQUIREMENTS FOR APPLICATION 15

### 

### 2. PASSWORD VERIFICATION USING PHP AND

### XAMPP

OVERVIEW **16**

[FUNCTIONAL REQUIREMENTS](#_heading=h.1t3h5sf)

* + 1. COMPONENTS **17**

OVERVIEW OF COMPONENTS **18**

BACKEND **28**

**3. SUMMARY AND CONCLUSIONS**

SUMMARY  **29**

CONCLUSIONS  **29**

**4**. [**REFERENCES**](#_heading=h.4d34og8)

[SOURCE CODE](#_heading=h.17dp8vu)  **30**

# ABSTRACT

Password verification system is very necessary for websites which has user registration feature because in case when user forgot his password and unable to login and then this password reset system helps to reset the password by sending link to his registered email address. With the help of PHPMAILER we can send a mail to the users when the new is registered to the account the mail has been sent to that user’s email id to verify the user’s Email id.

if the user forgot their password also, we can receive the reset link through mail to reset our password. Some of the software’s fails to send a mail to the customer to reset their password.

XAMPP is one of the to run the localhost in our system. It is simply a local host or server. This local server runs on your personal computer, whether it’s a desktop or a laptop. It is used to test clients or websites before publishing them to a remote web server.

On a local computer, the xampp server software provides a suitable environment for testing MYSQL, PHP, Apache, and Perl projects. Because most real-world-web server deployments share the same components as XAMPP, moving from a local test server to a live server is straightforward.

Composer is a package manager for PHP with Composer we can install the third-party libraries automatically. Composer would make the installation of third-party libraries much easier and faster.

After the installation is completed the **composer.json** file would be created on your folder. To use composer to install php packages, there are two options. you can create composer.json file, and use the Composer update command.

### 

### LIST OF FIGURES

**FIGURE No. TITLE PAGE No.**

1.1 Login page 18

1.2 Sign up Page 19

1.3 Forgotten password

page 20

1.4 Reset password page 21

1.5 Welcome page 22

1.6 Verification link page 23

2 Xampp 24

3 Phpmailer 27

4 MySQL connection 28

### 

### LIST OF ABBREVIATIONS

App - Application

HTML - Hyper Text Markup Language

CSS - Cascading Style Sheets

PHP - Hypertext Pre-processor

MYSQL - Structured Query Language

### INTRODUCTION

### What is Password verification system?

Password authentication refers to an authentication method that does not require a password or any other knowledge-based secret to log a user into a system. This method of authentication adopts the means of authorizing users by sending a unique, one-time code (OTP) or magic link to the user via email, push notification, or text message.

In this article, you'll learn how the infrastructure underlying password authentication works, and create a small app that uses this authentication approach.

## **Benefits:**

* **Improved users experience:**

This mode of authentication relieves users of the need to fill out password and confirm password fields, resulting in an increase in signup rate. It also alleviates the challenge of users having to remember complex passwords that adhere to various security policies.

* **Better security:**

Most internet users used the same password for multiple accounts before - and even after - the introduction of password generators and password managers. Additionally; gaining access to the user's password gives an attacker access to all of their accounts. Because there is no password to hack, password less authentication prevents this.

**Drawbacks:**

* **A single point of failure:**

Because it requires a verification medium (email, push notification, or text message), the user will be unable to access the system if they do not have access to one of these mediums.

### 

### Role of HTML, CSS, PHP & MYSQL:

### 

### A web page can contain headings, paragraphs, images, videos, and many other types of data. Front end developers use the HTML element to specify what kind of information each item on a webpage contains — for instance, the “p” HTML element indicates a paragraph. Developers also write HTML language code to specify how different items relate to one another in the page’s overall structure or document structure.

### CSS is used for defining the styles for web pages. It describes the look and formatting of a document which is written in a markup language. It provides an additional feature to HTML. ... CSS helps us to control the text colour, font style, the spacing between paragraphs, sizing of columns, layout designs, and many more. CSS provides these capabilities through a technical model based on a hierarchical scope of effect, the separation of style from content, and a well-defined set of published standards.

* PHP (Hypertext Pre-processor) is known as a general-purpose scripting language that can be used to develop dynamic and interactive websites. It was among the first server-side languages that could be embedded into HTML, making it easier to add functionality to web pages without needing to call external files for data.
* MYSQL database helps to automate data retrieving and provide great support in PHP MYSQL web application development.

### 

### Requirements of Application:

### Create a workspace with VS Code

### Design the general layout with HTML & CSS

### Create the required components with PHP.

### A database should be made in MYSQL for the backend purpose.

### Instead of changing in code, admin can make changes in the database, so that the change gets reflected in the website also.

**PASSWORD VERIFICATION USING PHP AND XAMPP**

**Overview:**

This project aims to verify if the given password is right or wrong. If the user has forgotten their password, they can update their own password by clicking the forgot password button at the bottom of the page .it will ask to enter the email id of your current account.so that by giving the current email id the admin will send you a verification link through user’s mail id. User can update the password after clicking link and no one can change the user’s password without their verification link sent by the admin.

Only the admin will be able to update the user’s details and passwords. There are many components like schedule, accommodation etc. This website is done with HTML, CSS, PHP and MYSQL.

This website is very easy to use for the users to register a new account or to change a password. Nowadays, many of the websites are using the HTML and CSS for their front end to show the website more attract to the users. When coming to the backend the PHP mailer is the best software to send a mails to the user’s without having any trouble. PHP my local is a website used to create database tables to store the information and details about user’s.

**Functional Requirements:**

To build this application, the main thing needed is the components where those processes occur. The components used in this application are listed below:

**COMPONENTS:**

* Login page
* Signup page
* Forgotten password page
* Reset password page
* welcome page
* Verification link page
* Xampp
* PhpMailer
* Backend

**OVERVIEW OF COMPONENTS:**

* **login page:**

login page where we can give a login credentials of email and password to Login

into the account

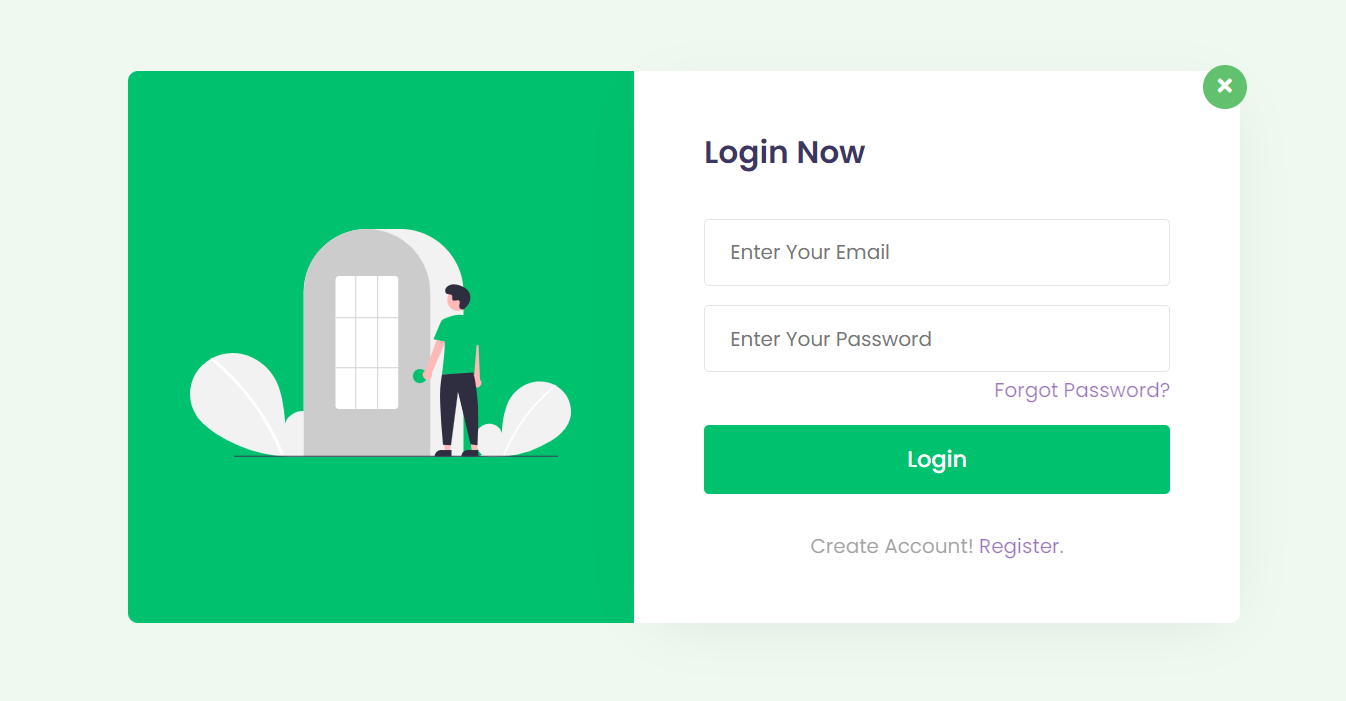


Fig 1.1: Login Page

* **Sign up page:**

Sign up page we can see that user can Register by filling the details like

name, email, password, repeat password to open a new account.

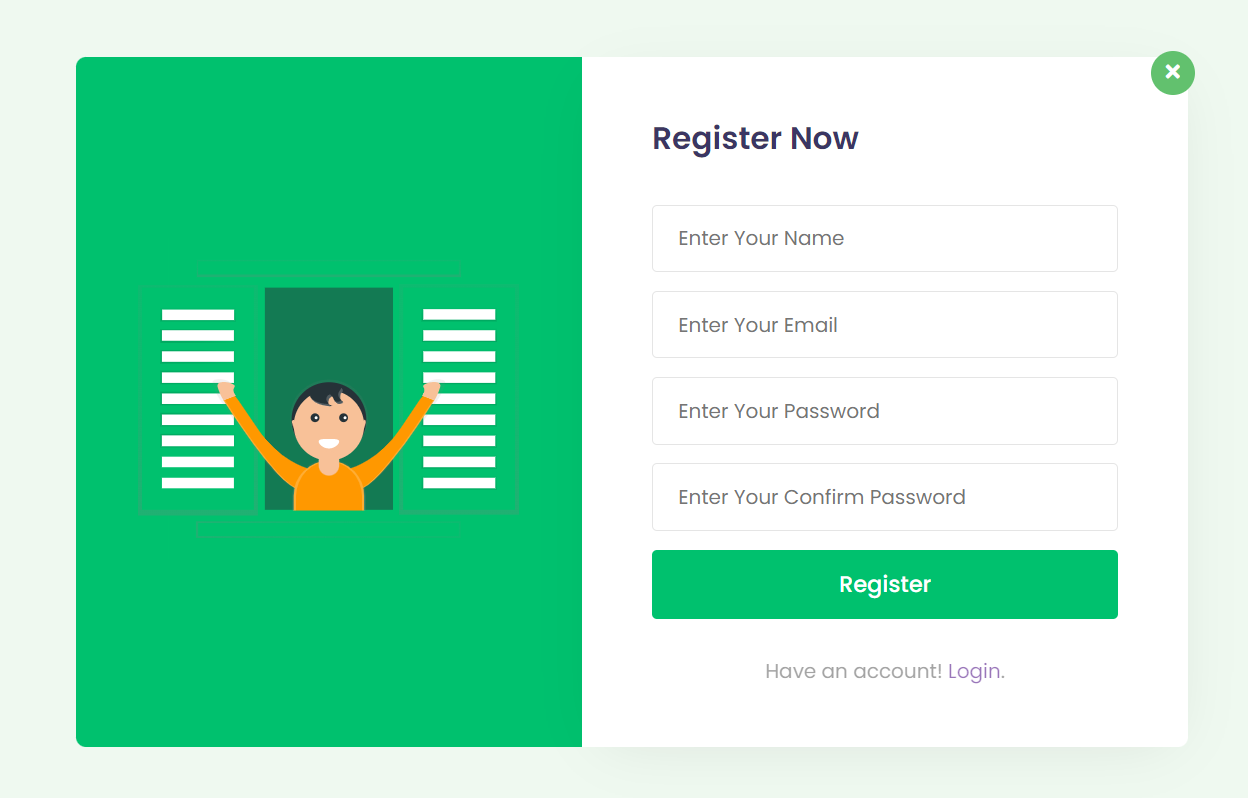


Fig 1.2: Register Page

* **Forgot Password page:**

in this we can see that when the user forgot their password. by giving the Email id to it the user can receive a link through mail to reset the new password.

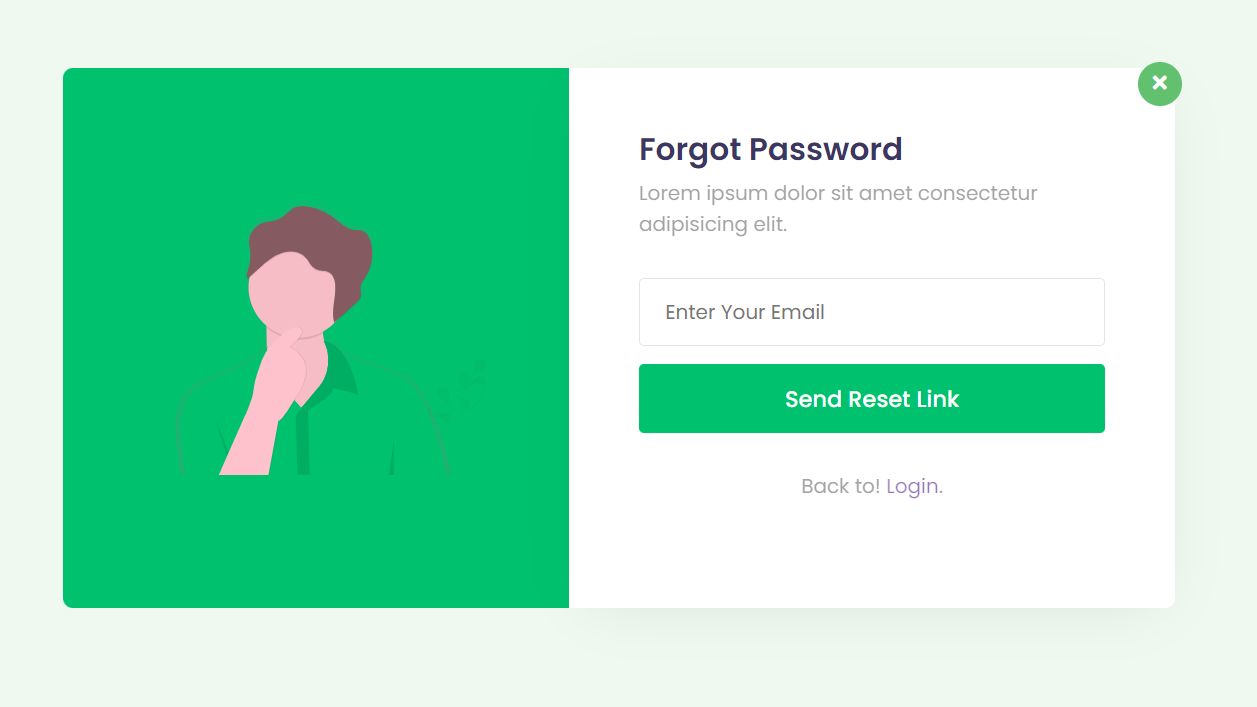


Fig 1.3: Forgot Page

* **Reset Password page:**

In this reset password page, we can see that after receiving the link through

user’s mail by clicking the link it will take to the reset password page.in this page we

can generate the new password.

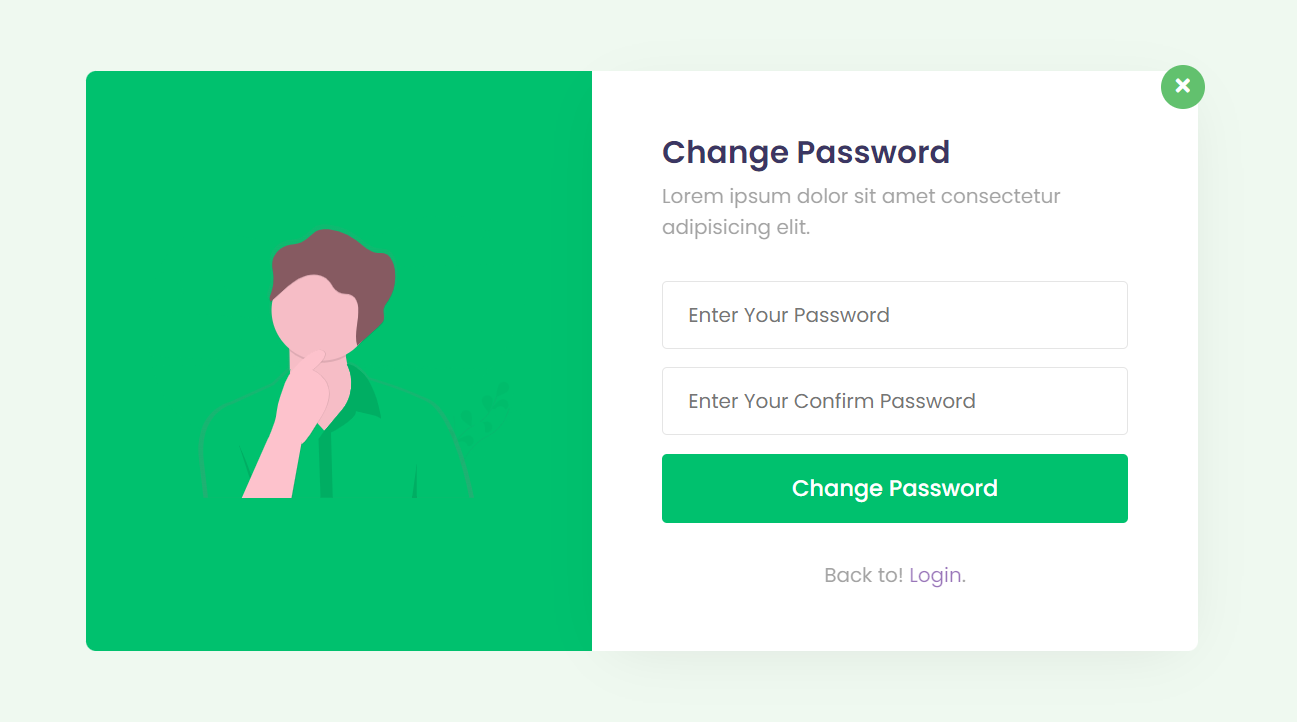


Fig 1.3: Contact Us Page

8

Fig 1.4: reset password link

* **Welcome Page:**



Fig 1.5: Welcome page

* **Verification link Page:**

In this page, the user will get the verification link through email by clicking on that link it will take to the change password page.

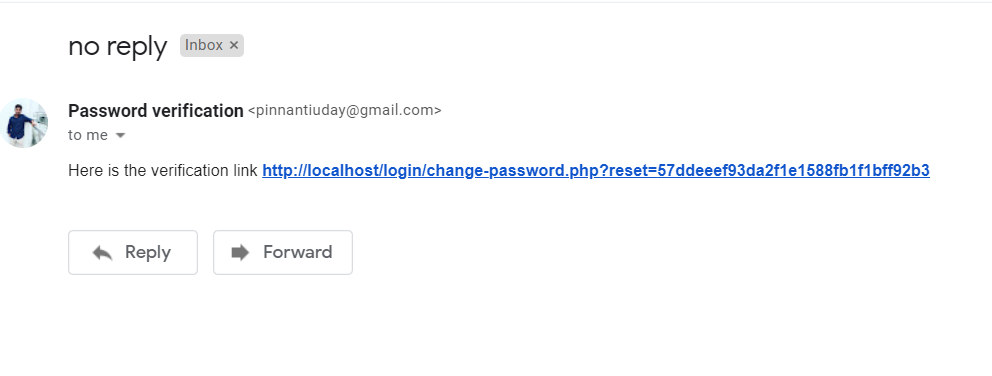


Fig 1.5: verification Page

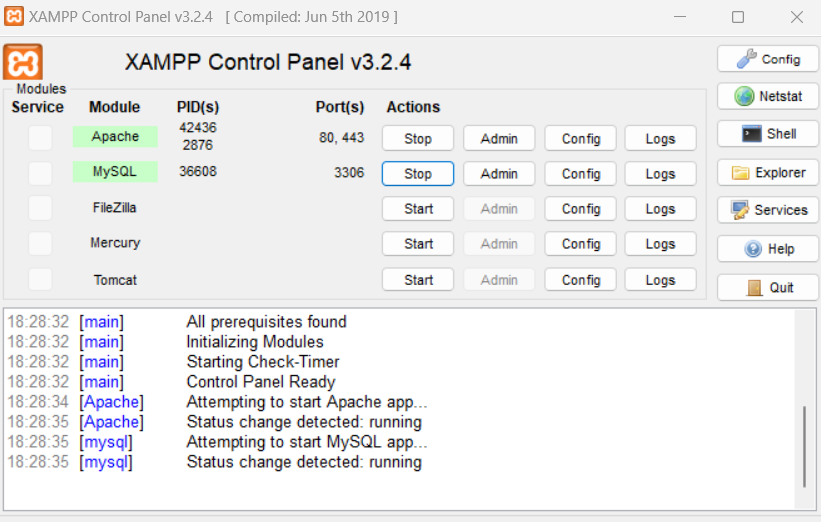
* **XAMPP:**

XAMPP is a cross-platform web server that is free and open-source.

XAMPP is a short form for Cross-Platform, Apache, MySQL, PHP, and Perl. XAMPP is a popular cross-platform web server that allows programmers to write and test their code on a local webserver. It was created by Apache Friends, and the public can revise or modify its native source code. It includes MariaDB, Apache HTTP Server, and interpreters for PHP and Perl, among other computer languages. Because of XAMPP’s simplicity of deployment, a developer can quickly and easily install a WAMP or LAMP stack on an operating system, with the added benefit that common add-in apps like WordPress and Joomla can also be loaded.

**Need for a XAMPP:**

* XAMPP is simply a local host or server.
* This local server runs on your personal computer, whether it's a desktop or a laptop.
* It is used to test clients on websites before publishing them to a   
  remote web server.
* On a local computer the XAMPP server software provides a suitable environment for testing MYSQL, PHP, Apache, and Perl projects. Because most real-world web servers to a live server are straightforward.



**Advantages and Disadvantages of XAMPP:**

* **Advantages:**
* In comparison to other web servers such as WAMP, it is simple to set up.
* It is Multi Cross-Platform, which implies it works on both Windows and Linux.
* With a single command, you may start and stop the entire web server and database stack.
* Both a full and a standard version of XAMPP are available.
* It has a control panel that you can see contains start and stop buttons for specific mechanisms, such as Apache, which is running through its Control panel.
* It also includes OpenSSL, phpMyAdmin, Media Wiki, Joomla, WordPress, and a lot of additional modules.
* **Disadvantages:**
* In comparison to the WAMP server, configuration and setting are more difficult.

**Components of XAMPP**

The components that are included in the XAMPP are:

**Cross-Platform:** Difference operating systems are installed in separate configurations on different local systems. The cross-platform component has been included to improve the functionality and reach to the Apache distribution package. It works with a variety of platforms, Including Windows, Linus, and MAC OS packages.

**Apache:** Apache is a cross-platform HTTP web server. It is used to transport web materials all over the world. If someone requests files, photos, or documents using their browser, HTTP server will serve such assets to clients.

**MariaDB Database:** XAMPP used to include MySQL DBMS; however, MariaDB has now taken its place. MYSQL is one of the most extensively used relational database management systems. It provides data storage, manipulation, retrieval, management, and deletion services via the internet.

**PHP:** The full form of PHP is Hypertext Pre-processor. PHP is a backend programming language and is most commonly used in web development. Users can use PHP to build dynamic websites and applications. It supports a variety of database management systems and may be installed on any platform. It was written in the c programming language.

**phpMyAdmin:** It is a database administration tool for MariaDB.

**XAMPP Control Panel:** The XAMPP Control Panel is a panel that assists in the operation and regulation of other XAMPP components.

**PHPMAILER:**

PHP Maileris the classic email sending library for PHP. It supports several ways of sending email messages such as mail (), Send mail, gmail, 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

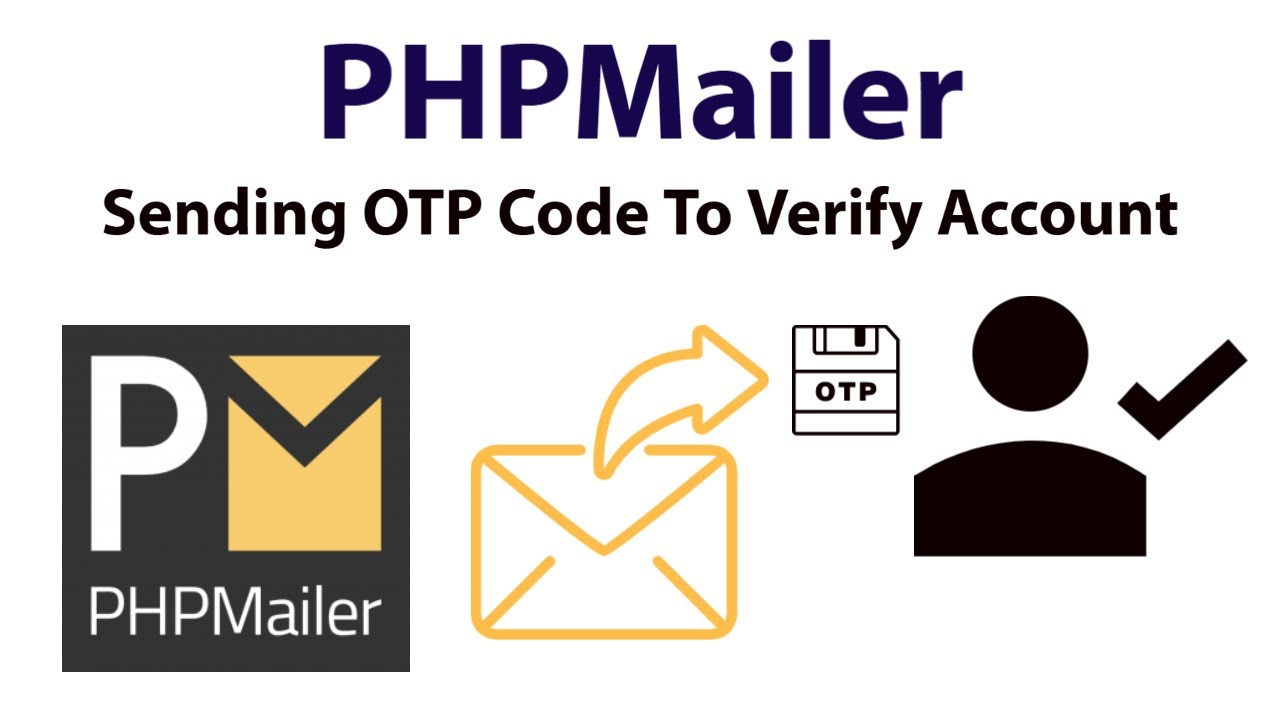


Fig 3: PHPmailer

* **BACKEND:**

The back-end duties involve the MYSQL connection and a service. For the back-end actions, a PHP document is done in the back end and thereby the MYSQL database is connected to the application.

For this to be carried out first of all a MySQL database should be created and thereby it should be connected with the back end through the PHP code via connection link.

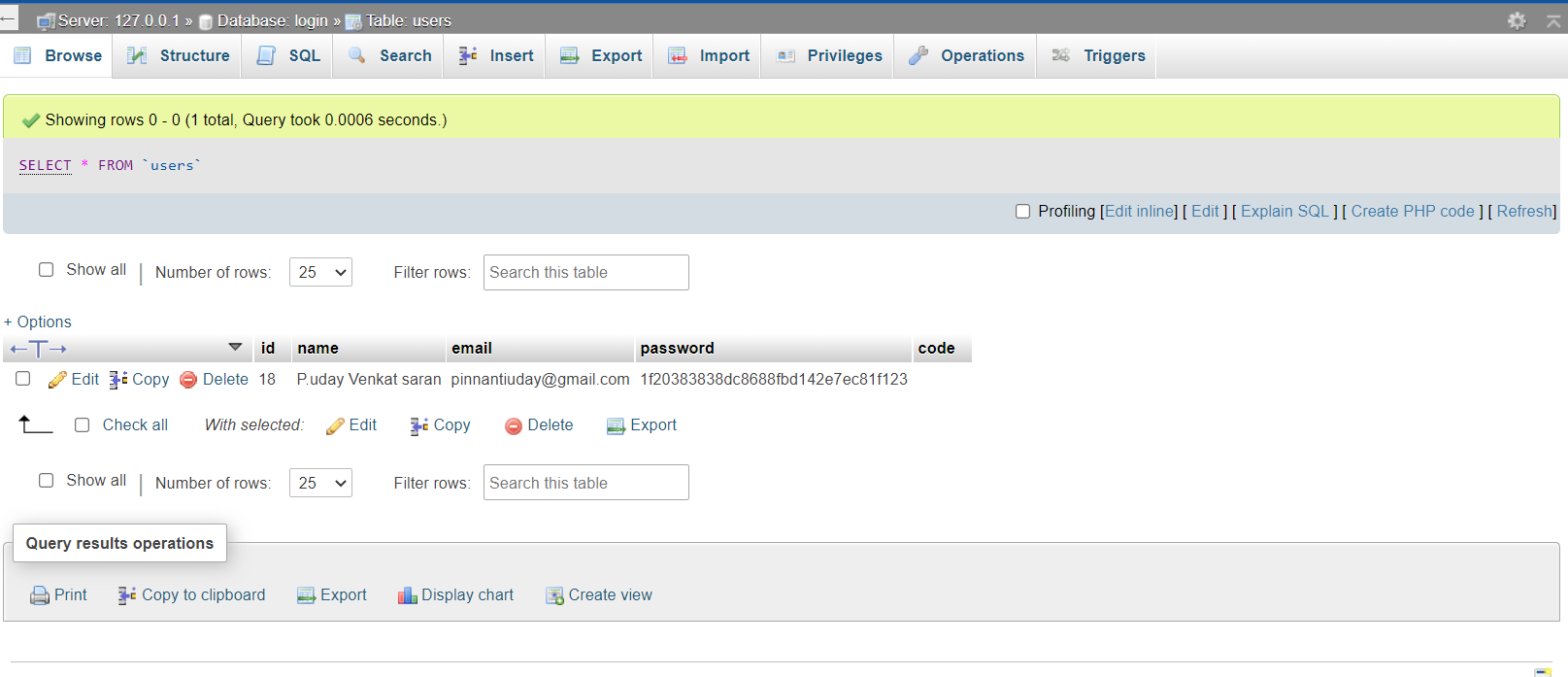


Fig 4: MYSQL Connection

**SUMMARY AND CONCLUSIONS**

**SUMMARY:**

Don’t save the password as plain text in the database table. If the database is compromised or hacked, then the hackers can easily access the individual account using the passwords.

If we hash the password and store it in the table, the hackers cannot access the individual account even though the database is hacked. Why because the hashed passwords are very hard to unpack.

So, if the database is hacked, the hackers cannot access the individual account. They are only able to see the information, not able to modify it.

So, I recommend all developers use password hashing while storing the password in the table.

**CONCLUSION:**

* The front-end languages such as HTML, CSS are becoming one among the emerging languages.
* This website is connected to the backend and thus collecting the total details about the journey becomes very easy.
* In today’s world, web development technology is an emerging technique which helps in many real time applications and saves people time and efforts.

**REFERENCES:**

[1] PHP Documentation

https://www.php.net/docs.php

[2] HTML & CSS Documentation

https://developer.mozilla.org/en-US/

[3] Bootstrap

<https://getbootstrap.com/>

[4] phpmyadmin

http://localhost/phpmyadmin/

**APPENDIX**

1. SOURCE CODE

**Index page:**

<?php

session\_start();

if (isset($\_SESSION['SESSION\_EMAIL'])) {

header("Location: welcome.php");

die();

}

include 'config.php';

$msg = "";

if (isset($\_GET['verification'])) {

if (mysqli\_num\_rows(mysqli\_query($conn, "SELECT \* FROM users WHERE code='{$\_GET['verification']}'")) > 0) {

$query = mysqli\_query($conn, "UPDATE users SET code='' WHERE code='{$\_GET['verification']}'");

if ($query) {

$msg = "<div class='alert alert-success'>Account verification has been successfully completed.</div>";

}

} else {

header("Location: index.php");

}

}

if (isset($\_POST['submit'])) {

$email = mysqli\_real\_escape\_string($conn, $\_POST['email']);

$password = mysqli\_real\_escape\_string($conn, md5($\_POST['password']));

$sql = "SELECT \* FROM users WHERE email='{$email}' AND password='{$password}'";

$result = mysqli\_query($conn, $sql);

if (mysqli\_num\_rows($result) === 1) {

$row = mysqli\_fetch\_assoc($result);

if (empty($row['code'])) {

$\_SESSION['SESSION\_EMAIL'] = $email;

header("Location: welcome.php");

} else {

$msg = "<div class='alert alert-info'>First verify your account and try again.</div>";

}

} else {

$msg = "<div class='alert alert-danger'>Email or password do not match.</div>";

}

}

?>

<!DOCTYPE html>

<html lang="zxx">

<head>

<title>Password verification - uday</title>

<!-- Meta tag Keywords -->

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta charset="UTF-8" />

<meta name="keywords"

content="Login Form" />

<!-- //Meta tag Keywords -->

<link href="//fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display=swap" rel="stylesheet">

<!--/Style-CSS -->

<link rel="stylesheet" href="css/style.css" type="text/css" media="all" />

<!--//Style-CSS -->

<script src="https://kit.fontawesome.com/af562a2a63.js" crossorigin="anonymous"></script>

</head>

<body>

<!-- form section start -->

<section class="w3l-mockup-form">

<div class="container">

<!-- /form -->

<div class="workinghny-form-grid">

<div class="main-mockup">

<div class="alert-close">

<span class="fa fa-close"></span>

</div>

<div class="w3l\_form align-self">

<div class="left\_grid\_info">

<img src="images/image.svg" alt="">

</div>

</div>

<div class="content-wthree">

<h2>Login Now</h2>

<p></p>

<?php echo $msg; ?>

<form action="" method="post">

<input type="email" class="email" name="email" placeholder="Enter Your Email" required>

<input type="password" class="password" name="password" placeholder="Enter Your Password" style="margin-bottom: 2px;" required>

<p><a href="forgot-password.php" style="margin-bottom: 15px; display: block; text-align: right;">Forgot Password?</a></p>

<button name="submit" name="submit" class="btn" type="submit">Login</button>

</form>

<div class="social-icons">

<p>Create an Account! <a href="register.php">Register</a></a></a></a></a></a></a></a>.</p>

</div>

</div>

</div>

</div>

<!-- //form -->

</div>

</section>

<!-- //form section start -->

<script src="js/jquery.min.js"></script>

<script>

$(document).ready(function (c) {

$('.alert-close').on('click', function (c) {

$('.main-mockup').fadeOut('slow', function (c) {

$('.main-mockup').remove();

});

});

});

</script>

</body>

</html>

**Register page:**

<?php

//Import PHPMailer classes into the global namespace

//These must be at the top of your script, not inside a function

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\SMTP;

use PHPMailer\PHPMailer\Exception;

session\_start();

if (isset($\_SESSION['SESSION\_EMAIL'])) {

header("Location: welcome.php");

die();

}

//Load Composer's autoloader

require 'vendor/autoload.php';

include 'config.php';

$msg = "";

if (isset($\_POST['submit'])) {

$name = mysqli\_real\_escape\_string($conn, $\_POST['name']);

$email = mysqli\_real\_escape\_string($conn, $\_POST['email']);

$password = mysqli\_real\_escape\_string($conn, md5($\_POST['password']));

$confirm\_password = mysqli\_real\_escape\_string($conn, md5($\_POST['confirm-password']));

$code = mysqli\_real\_escape\_string($conn, md5(rand()));

if (mysqli\_num\_rows(mysqli\_query($conn, "SELECT \* FROM users WHERE email='{$email}'")) > 0) {

$msg = "<div class='alert alert-danger'>{$email} - This email address has been already exists.</div>";

} else {

if ($password === $confirm\_password) {

$sql = "INSERT INTO users (name, email, password, code) VALUES ('{$name}', '{$email}', '{$password}', '{$code}')";

$result = mysqli\_query($conn, $sql);

if ($result) {

echo "<div style='display: none;'>";

//Create an instance; passing `true` enables exceptions

$mail = new PHPMailer(true);

try {

//Server settings

$mail->SMTPDebug = SMTP::DEBUG\_SERVER; //Enable verbose debug output

$mail->isSMTP(); //Send using SMTP

$mail->Host = 'smtp.gmail.com'; //Set the SMTP server to send through

$mail->SMTPAuth = true; //Enable SMTP authentication

$mail->Username = "pinnantiuday@gmail.com"; //SMTP username

$mail->Password = "aqfodrshwiyeltjc"; //SMTP password

$mail->SMTPSecure = PHPMailer::ENCRYPTION\_SMTPS; //Enable implicit TLS encryption

$mail->Port = 465; //TCP port to connect to; use 587 if you have set `SMTPSecure = PHPMailer::ENCRYPTION\_STARTTLS`

//Recipients

$mail->setFrom("pinnantiuday@gmail.com","Password verification");

$mail->addAddress($email);

//Content

$mail->isHTML(true); //Set email format to HTML

$mail->Subject = 'no reply';

$mail->Body = 'Here is the verification link <b><a href="http://localhost/login/?verification='.$code.'">http://localhost/login/?verification='.$code.'</a></b>';

$mail->send();

echo 'Message has been sent';

} catch (Exception $e) {

echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";

}

echo "</div>";

$msg = "<div class='alert alert-info'>We've send a verification link on your email address.</div>";

} else {

$msg = "<div class='alert alert-danger'>Something wrong went.</div>";

}

} else {

$msg = "<div class='alert alert-danger'>Password and Confirm Password do not match</div>";

}

}

}

?>

<!DOCTYPE html>

<html lang="zxx">

<head>

<title>Password verification - uday</title>

<!-- Meta tag Keywords -->

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta charset="UTF-8" />

<meta name="keywords"

content="Login Form" />

<!-- //Meta tag Keywords -->

<link href="//fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display=swap" rel="stylesheet">

<!--/Style-CSS -->

<link rel="stylesheet" href="css/style.css" type="text/css" media="all" />

<!--//Style-CSS -->

<script src="https://kit.fontawesome.com/af562a2a63.js" crossorigin="anonymous"></script>

</head>

<body>

<!-- form section start -->

<section class="w3l-mockup-form">

<div class="container">

<!-- /form -->

<div class="workinghny-form-grid">

<div class="main-mockup">

<div class="alert-close">

<span class="fa fa-close"></span>

</div>

<div class="w3l\_form align-self">

<div class="left\_grid\_info">

<img src="images/image2.svg" alt="">

</div>

</div>

<div class="content-wthree">

<h2>Register Now</h2>

<p></p>

<?php echo $msg; ?>

<form action="" method="post">

<input type="text" class="name" name="name" placeholder="Enter Your Name" value="<?php if (isset($\_POST['submit'])) { echo $name; } ?>" required>

<input type="email" class="email" name="email" placeholder="Enter Your Email" value="<?php if (isset($\_POST['submit'])) { echo $email; } ?>" required>

<input type="password" class="password" name="password" placeholder="Enter Your Password" required>

<input type="password" class="confirm-password" name="confirm-password" placeholder="Enter Your Confirm Password" required>

<button name="submit" class="btn" type="submit">Register</button>

</form>

<div class="social-icons">

<p>Have an account! <a href="index.php">Login</a></a></a></a>.</p>

</div>

</div>

</div>

</div>

<!-- //form -->

</div>

</section>

<!-- //form section start -->

<script src="js/jquery.min.js"></script>

<script>

$(document).ready(function (c) {

$('.alert-close').on('click', function (c) {

$('.main-mockup').fadeOut('slow', function (c) {

$('.main-mockup').remove();

});

});

});

</script>

</body>

</html>

**Welcome page:**

<!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>Document</title>

<style>

body{

background-image: url(https://wallpapertag.com/wallpaper/full/8/0/b/115326-background-white-1920x1138-for-xiaomi.jpg);

height: 350px;

width: 920px;

background-repeat: no-repeat;

background-position: center;

margin-left: 300px;

margin-right: 400px;

margin-top: 150px;

padding-left: 20px;

}

button {

padding: 10px;

width: 140%;

border-radius: 15px;

background-color: white;

cursor: pointer;

font-size: 15px;

text-align: center;

}

.btn {

text-align: left;

margin-left: 850px;

position: absolute;

top: 8px;

right: 60px;

font-size: 18px;

}

.venkat{

text-align: center;

font-size: 50px;

position: relative;

overflow: hidden;

width: 900px;

height: 100px;

}

#slide {

position: relative;

right: 1200px;

-webkit-animation: slide 0.5s forwards;

-webkit-animation-delay: 0s;

animation: slide 0.5s forwards;

animation-delay: 0.5s;

}

@-webkit-keyframes slide {

100% { right: 0; }

}

@keyframes slide {

100% { right: 0; }

}

</style>

</head>

<body>

<div class="btn">

<br><br>

<a href="logout.php">

<button type="submit">Logout</button>

</a>

</div>

<br><br><br>

<div class="venkat" id="slide">

<?php

session\_start();

if (!isset($\_SESSION['SESSION\_EMAIL'])) {

header("Location: index.php");

die();

}

include 'config.php';

$query = mysqli\_query($conn, "SELECT \* FROM users WHERE email='{$\_SESSION['SESSION\_EMAIL']}'");

if (mysqli\_num\_rows($query) > 0) {

$row = mysqli\_fetch\_assoc($query);

echo "Welcome " . $row['name'];

}

?>

</div>

<h2>

</h2>

</body>

</html>

**Logout page:**

<?php

session\_start();

session\_unset();

session\_destroy();

header("Location: index.php");

**Forgot-Password page :**

<?php

session\_start();

if (isset($\_SESSION['SESSION\_EMAIL'])) {

header("Location: welcome.php");

die();

}

//Import PHPMailer classes into the global namespace

//These must be at the top of your script, not inside a function

use PHPMailer\PHPMailer\PHPMailer;

use PHPMailer\PHPMailer\SMTP;

use PHPMailer\PHPMailer\Exception;

//Load Composer's autoloader

require 'vendor/autoload.php';

include 'config.php';

$msg = "";

if (isset($\_POST['submit'])) {

$email = mysqli\_real\_escape\_string($conn, $\_POST['email']);

$code = mysqli\_real\_escape\_string($conn, md5(rand()));

if (mysqli\_num\_rows(mysqli\_query($conn, "SELECT \* FROM users WHERE email='{$email}'")) > 0) {

$query = mysqli\_query($conn, "UPDATE users SET code='{$code}' WHERE email='{$email}'");

if ($query) {

echo "<div style='display: none;'>";

//Create an instance; passing `true` enables exceptions

$mail = new PHPMailer(true);

try {

//Server settings

$mail->SMTPDebug = SMTP::DEBUG\_SERVER; //Enable verbose debug output

$mail->isSMTP(); //Send using SMTP

$mail->Host = 'smtp.gmail.com'; //Set the SMTP server to send through

$mail->SMTPAuth = true; //Enable SMTP authentication

$mail->Username = "pinnantiuday@gmail.com"; //SMTP username

$mail->Password = "aqfodrshwiyeltjc"; //SMTP password

$mail->SMTPSecure = PHPMailer::ENCRYPTION\_SMTPS; //Enable implicit TLS encryption

$mail->Port = 465; //TCP port to connect to; use 587 if you have set `SMTPSecure = PHPMailer::ENCRYPTION\_STARTTLS`

//Recipients

$mail->setFrom( "pinnantiuday@gmail.com","Password verification");

$mail->addAddress($email);

//Content

$mail->isHTML(true); //Set email format to HTML

$mail->Subject = 'no reply';

$mail->Body = 'Here is the verification link <b><a href="http://localhost/login/change-password.php?reset='.$code.'">http://localhost/login/change-password.php?reset='.$code.'</a></b>';

$mail->send();

echo 'Message has been sent';

} catch (Exception $e) {

echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";

}

echo "</div>";

$msg = "<div class='alert alert-info'>We've send a verification link on your email address.</div>";

}

} else {

$msg = "<div class='alert alert-danger'>$email - This email address do not found.</div>";

}

}

?>

<!DOCTYPE html>

<html lang="zxx">

<head>

<title>Password verification - uday</title>

<!-- Meta tag Keywords -->

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta charset="UTF-8" />

<meta name="keywords"

content="Login Form" />

<!-- //Meta tag Keywords -->

<link href="//fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display=swap" rel="stylesheet">

<!--/Style-CSS -->

<link rel="stylesheet" href="css/style.css" type="text/css" media="all" />

<!--//Style-CSS -->

<script src="https://kit.fontawesome.com/af562a2a63.js" crossorigin="anonymous"></script>

</head>

<body>

<!-- form section start -->

<section class="w3l-mockup-form">

<div class="container">

<!-- /form -->

<div class="workinghny-form-grid">

<div class="main-mockup">

<div class="alert-close">

<span class="fa fa-close"></span>

</div>

<div class="w3l\_form align-self">

<div class="left\_grid\_info">

<img src="images/image3.svg" alt="">

</div>

</div>

<div class="content-wthree">

<h2>Forgot Password</h2>

<p>Lorem ipsum dolor sit amet consectetur adipiscing elit. </p>

<?php echo $msg; ?>

<form action="" method="post">

<input type="email" class="email" name="email" placeholder="Enter Your Email" required>

<button name="submit" class="btn" type="submit">Send Reset Link</button>

</form>

<div class="social-icons">

<p>Back to! <a href="index.php">Login</a></a></a></a>.</p>

</div>

</div>

</div>

</div>

<!-- //form -->

</div>

</section>

<!-- //form section start -->

<script src="js/jquery.min.js"></script>

<script>

$(document).ready(function (c) {

$('.alert-close').on('click', function (c) {

$('.main-mockup').fadeOut('slow', function (c) {

$('.main-mockup').remove();

});

});

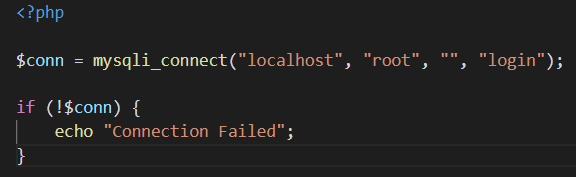
});

</script>

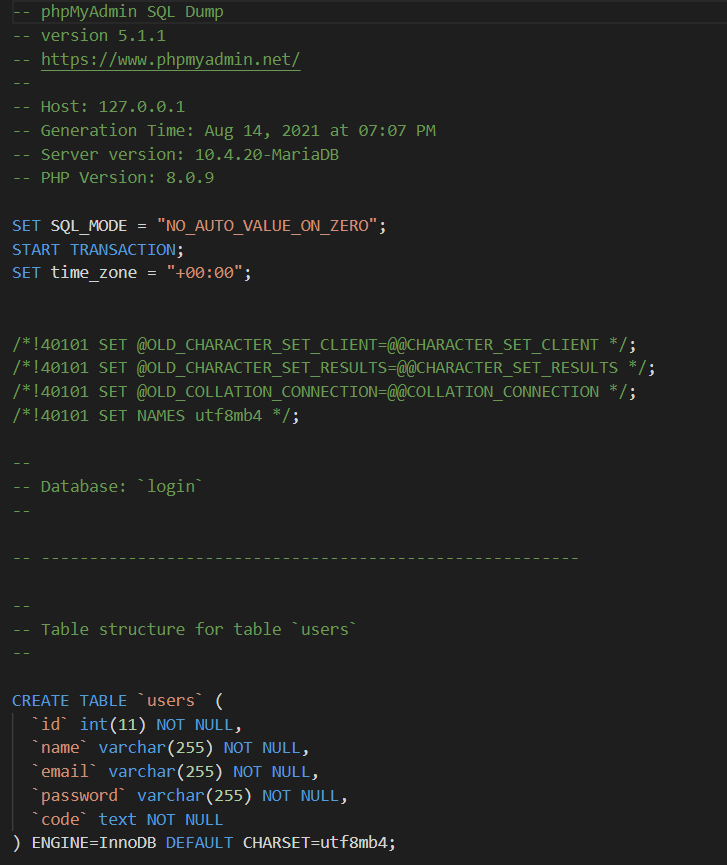
</body>

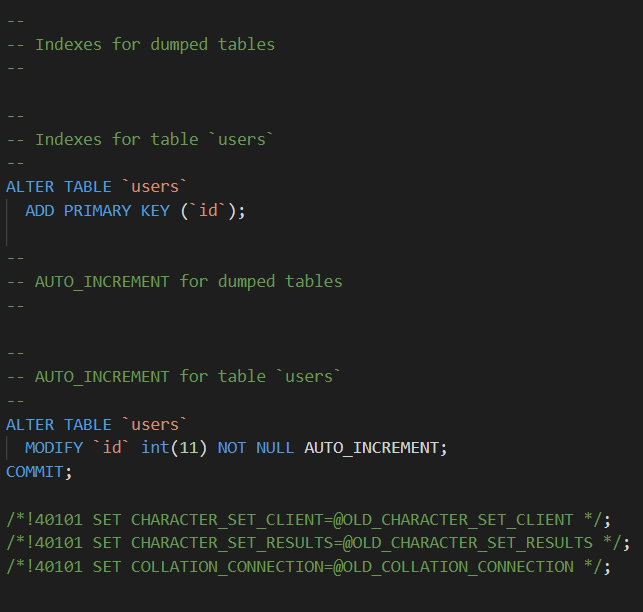
</html>

**Config.php page:**



**Login.sql page:**

****



**Change-password.php page:**

