



Bachelor of Information Technology (Hons) Assignment Cover Sheet

Course Code: EC3312 Course Title: Information Assurance Security
Assignment Title: Assignment I Due Date: 10th August, 2024
Date Submitted: 10th August, 2024 Lecturer Name: Mr. Sabin Poudel

To be completed if this is an individual assignment

I declare that this assignment is my individual work. I have not worked collaboratively nor have I copied from any other student's work or from any other source except where due acknowledgement is made explicitly in the text, nor has any part been written for me by another person.

Student Name: Kishor Chaudhary Student ID: 00020049
Signature: Kishor

To be completed if this is a group assignment

We declare that this is a group assignment and that no part of this submission has been copied from any other student's work or from any other source except where due acknowledgement is made explicitly in the text, nor has any part been written for us by another person.

Student ID	Student Name	Signature
_____	_____	_____
_____	_____	_____
_____	_____	_____
_____	_____	_____

Lecturer's comments: _____

Total Marks: _____ Lecturer's Signature: _____

Feedback to Student:

I/We acknowledged receiving feedback from the lecturer on this assignment.

Student's Signature: _____

Extension certification:

This assignment has been given an extension and is now due on _____.

Lecturer's Signature: _____

Contents

Introduction	3
Objective	3
Workflow	4
Code.....	5
Code for register.php	5
Code for login.php page.....	7
Code for opt.php page	9
Code for password.php page	11
Code for index.php page	12
Code for logout.php page	13
Screenshot.....	26
Database Design	37
Conclusion.....	38

Introduction

PHP(Hypertext Preprocessor) is a widely-used, open-source scripting language primarily used for web development. It is embedded within HTML(Hypertext Markup Language) code, allowing for dynamic content generation.

Key feature of PHP includes:

- Server-side scripting
- Database connectivity
- Form handling
- Session management
- Error reporting

OTP stands for One-Time Password is a security measure designed to enhance the protection of sensitive information. It is used in login/registration and sending fund for security.

Objective

The objective of making this report to create a login/ registration form with the PHP and generating OTP and sending to email for more secure and only authorized person can login to the system. With the help of this system there will be stronger authentication, protection against credential theft, user confidence and reduced fraud.

Workflow

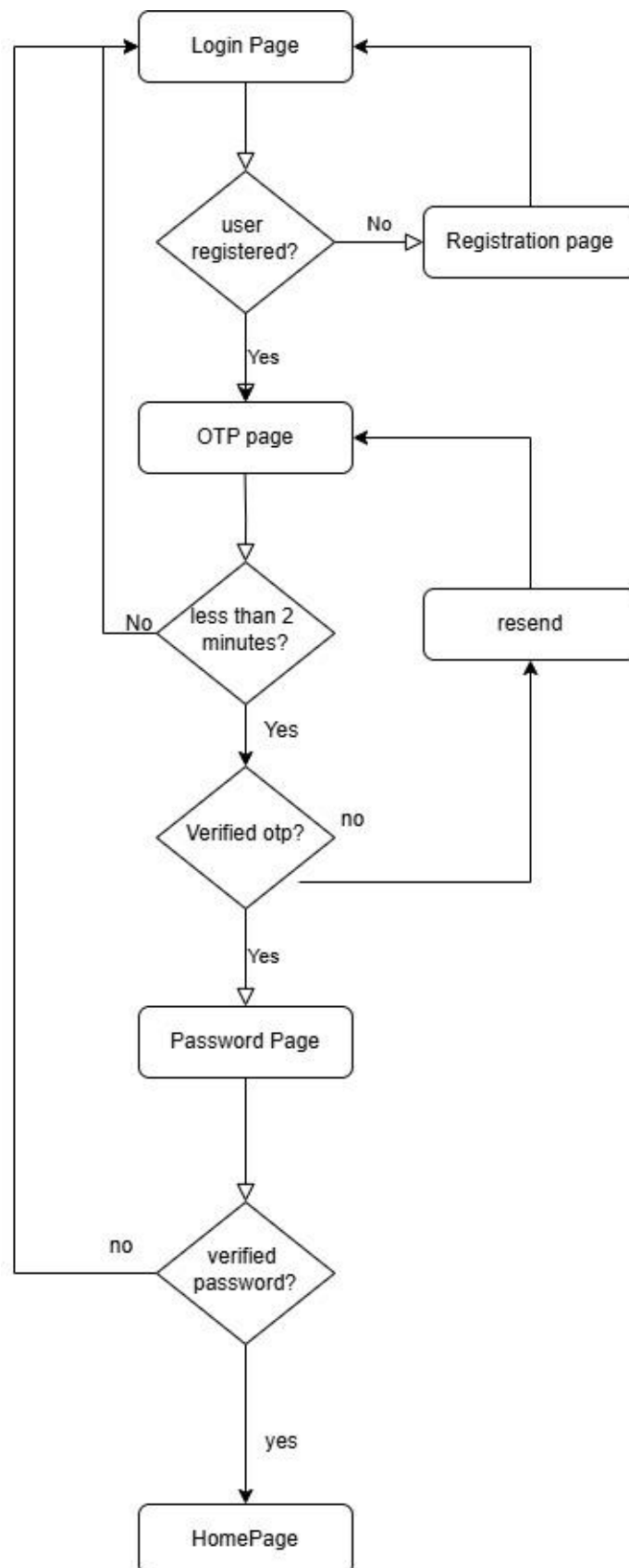


Fig. Workflow diagram of the system

Code

Code for register.php

```
<?php
@include 'config.php';
if (isset($_POST['submit'])) {

    $name = mysqli_real_escape_string($conn, $_POST['name']);
    $email = mysqli_real_escape_string($conn, $_POST['email']);
    $pass = mysqli_real_escape_string($conn, md5($_POST['pass']));
    $cpass = mysqli_real_escape_string($conn, md5($_POST['cpass']));
    $created_at = date('Y-m-d H:i:s');

    if ($pass != $cpass) {
        echo "<script>alert('Passwords do not match');</script>";
    } else {

        $check_email = mysqli_query($conn, "SELECT email FROM users WHERE email = '$email'");

        if (mysqli_num_rows($check_email) > 0) {
            echo "<script>alert('Email already registered!');</script>";
        } else {

            $insert = "INSERT INTO users(name, email, password, created_at) VALUES('$name', '$email', '$pass', '$created_at')";

            if (mysqli_query($conn, $insert)) {
                echo "<script>alert('Registration successful!');</script>";
            } else {

                echo"<script>alert('Connection Error');</script>";
            }
        }
    }
}

?>

<!DOCTYPE html>
<html lang="en">
```

```

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Register | Signup</title>
  <!-- style -->
  <link rel="stylesheet" href="assets/css/style.css">
  <!-- font awesome link -->
  <script src="https://kit.fontawesome.com/9361e3694f.js"
crossorigin="anonymous"> </script>
</head>
<body>

  <div class="register-container">
    <form method="post" action="register.php" onclick="return validateForm(event);" >
      <h3>Sign Up</h3>
      <div class="inputBox">
        <input type="text" name="name" id="name" required>
        <span>Username</span>
        <i></i>
      </div>
      <div class="inputBox">
        <input type="email" name="email" id="email" required>
        <span>Email</span>
        <i></i>
      </div>
      <div class="inputBox">
        <input type="password" name="pass" id="pass" required>
        <span>Enter Password</span>
        <i></i>
      </div>
      <div class="inputBox">
        <input type="password" name="cpass" id="cpass" required>
        <span>Confirm Password</span>
        <i></i>
      </div>
      <input type="submit" name="submit" value="Register">
      <div class="links">
        <a href="login.php">Already have an account? Sign In</a>
      </div>
    </form>
  </div>

  <script src="assets/js/script.js"> </script>
</body>
</html>

```

Code for login.php page

```
<?php
@include 'config.php';

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

require 'vendor/autoload.php';

function sendOTP($email, $otp)
{
    $mail = new PHPMailer(true);

    try {
        //Message sending setting using phpmailer
        $mail->isSMTP();
        $mail->Host      = 'smtp.gmail.com';
        $mail->SMTPAuth  = true;
        $mail->Username  = 'kishorc2000@gmail.com';
        $mail->Password  = 'gkct mujl rygp ygsk';
        $mail->SMTPSecure = 'tls';
        $mail->Port      = 587;

        $mail->setFrom('work.kishorc2000@gmail.com', 'Kishor');
        $mail->addAddress($email);

        // Message which will be sent to email
        $mail->isHTML(true);
        $mail->Subject = 'OTP Code Verification';
        $mail->Body    = "Your OTP code is <b>$otp</b>. It is valid for 2 minutes.";
        $mail->AltBody = "Your OTP code is $otp. It is valid for 2 minutes.";

        $mail->send();
        echo "<script>alert('OTP has been sent to your email.');
```

```

// Validate email format
if (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
    echo "<script>alert('Invalid email format');</script>";
    exit();
}

// Check if the email is registered
$check_email = mysqli_query($conn, "SELECT * FROM users WHERE email = '$email'
LIMIT 1");

if (mysqli_num_rows($check_email) > 0) {

    $otp =
substr(str_shuffle('0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz'), 0, 6);

    // Save the OTP to the database
    $query = mysqli_query($conn, "INSERT INTO otps (email, otp, created_at) VALUES
('$email', '$otp', NOW())");

    // Send OTP to email
    sendOTP($email, $otp);

    header("Location: otp.php?email=" . urlencode($email));
    exit();
} else {
    echo "<script>alert('Email is not registered!');</script>";
}
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login | Signup</title>
    <!-- style -->
    <link rel="stylesheet" href="assets/css/style.css">
    <!-- font awesome link -->
    <script src="https://kit.fontawesome.com/9361e3694f.js"
crossorigin="anonymous"></script>
</head>

```



```

<body>
  <div class="login-container">
    <form method="post" action="login.php" id="loginForm" onclick="return
validate();">
      <h3>Sign In</h3>
      <div class="inputBox">
        <input type="email" name="email" id="email" required>
        <span>Email</span>
        <i></i>
      </div>

      <input type="submit" name="submit" value="Get OTP">

      <div class="links">
        <a href="#">Forgot Password</a>
        <a href="register.php">Sign Up</a>
      </div>
    </form>
  </div>
  <script src="assets/js/script.js"> </script>
</body>
</html>

```

Code for opt.php page

```

<?php
@include 'config.php';

if (isset($_POST['submit'])) {
  $otp = mysqli_real_escape_string($conn, $_POST['otp']);
  $email = mysqli_real_escape_string($conn, $_POST['email']);

  $result = mysqli_query($conn, "SELECT * FROM otps WHERE email = '$email' AND otp = '$otp' ORDER BY created_at DESC LIMIT 1");

  if (mysqli_num_rows($result) > 0) {
    $row = mysqli_fetch_assoc($result);
    $otp_time = strtotime($row['created_at']);
    $current_time = time();

    if (($current_time - $otp_time) <= 120) {
      header("Location: password.php?email=" . urldecode($email));
      exit();
    } else {
      echo "<script>alert('OTP has expired!');</script>";
    }
  }
}

```

```

    }
}

else {
    echo "<script>alert('Invalid OTP!');</script>";
}
}
?>

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>OTP</title>
    <!-- style -->
    <link rel="stylesheet" href="assets/css/style.css">
    <!-- font awesome link -->
    <script src="https://kit.fontawesome.com/9361e3694f.js"
crossorigin="anonymous"> </script>
</head>

<body>
    <div class="otp-container">
        <form action="otp.php" method="post">
            <h3>OTP Conformation</h3>
            <div class="inputBox">
                <input type="text" id="otp" name="otp" required>
                <span>Enter OTP</span>
                <i></i>
            </div>
            <input type="hidden" name="email" value="<?php echo htmlspecialchars(
$_GET['email']); ?>">
            <input type="submit" name="submit" value="Verify OTP">
            <div class="links">
                <a href="login.php?submit=true&email=<?php urlencode( $_GET['email']);
?>">Resend</a>
                <a href="login.php">Go Back</a>
            </div>
        </form>
    </div>
</body>
</html>

```

Code for password.php page

```
<?php
@include 'config.php';

if (isset($_POST['submit'])) {
    $pass = mysqli_real_escape_string($conn, md5($_POST['pass']));
    $email = mysqli_real_escape_string($conn, $_POST['email']);

    // Check if the email and password are matched
    $query = "SELECT * FROM users WHERE email = '$email' AND password = '$pass'
LIMIT 1";
    $result = mysqli_query($conn, $query);

    if (mysqli_num_rows($result) > 0) {

        echo "<script>alert('Login successful! Redirecting....');</script>";

        header("Location: index.php");
        exit();
    } else {
        echo "<script>alert('Incorrect password');</script>";
    }
}
?>
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Enter Password</title>
    <link rel="stylesheet" href="assets/css/style.css">
</head>

<body>
    <div class="password-container">
        <form method="post" action="password.php">
            <h3>Enter Password</h3>
            <div class="inputBox">
                <input type="password" id="pass" name="pass" required>
                <span>Enter Password</span>
            </div>
        </form>
    </div>
</body>
</html>
```

```

        <input type="hidden" name="email" value="<?php echo htmlspecialchars(
$_GET['email']); ?>">
        <input type="submit" name="submit" value="Login">
        <div class="links">
            <a href="register.php">Sign Up</a>
            <a href="login.php">Go Back</a>
        </div>
    </form>
</div>
</body>

</html>

```

Code for index.php page

```

<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=">
    <title>Home</title>
    <!-- style -->
    <link rel="stylesheet" href="assets/css/style.css">
    <!-- font awesome link -->
    <script src="https://kit.fontawesome.com/9361e3694f.js"
crossorigin="anonymous"> </script>
</head>

<body>
    <div class="index-container">
        <h1>Welcome</h1>
        <a href="logout.php">logout</a>
    </div>

</body>

</html>

```

Code for logout.php page

```
<?php
```

```
@include 'config.php';
```

```
header('location:login.php');
```

Code for style.css

```
@import
```

```
url('https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display=swap');
```

```
:root {  
  --blue: #45f3ff;  
  --pink: #d9138a;  
  --black-bg: #353434;  
  --grey: #8f8f8f;  
  --light-color: #666;  
  --black: #23233a;  
  --bg:#1a1d25;  
}  
* {  
  margin: 0;  
  padding: 0;  
  box-sizing: border-box;  
  font-family: 'Poppins', sans-serif;  
}  
body {  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  min-height: 100vh;  
  background: var(--black-bg);  
  color: white;  
}
```

```
.login-container {  
  position: relative;  
  width: 370px;  
  height: 450px;  
  background: var(--black);  
  border-radius: 50px 5px;  
  overflow: hidden;
```

```

}
.login-container::before {
  content: "";
  position: absolute;
  top: -50%;
  left: -50%;
  width: 370px;
  height: 450px;
  background: linear-gradient(60deg, transparent, var(--blue), var(--blue));
  transform-origin: bottom right;
  animation: animate 6s linear infinite;
}
.login-container::after {
  content: "";
  position: absolute;
  top: -50%;
  left: -50%;
  width: 370px;
  height: 450px;
  background: linear-gradient(60deg, transparent, var(--pink), var(--pink));
  transform-origin: bottom right;
  animation: animate 6s linear infinite;
  animation-delay: -3s;
}

@keyframes animate {
  0% {
    transform: rotate(0deg);
  }
  100% {
    transform: rotate(360deg);
  }
}

.login-container form {
  position: absolute;
  inset: 2px;
  border-radius: 50px 5px;
  background: var(--bg);
  z-index: 10;
  padding: 30px 40px;
  display: flex;
  flex-direction: column;

```

```

}
.login-container h3 {
  color: var(--blue);
  font-size: 35px;
  font-weight: 500;
  text-align: center;
}
.login-container .inputBox {
  position: relative;
  width: 300px;
  margin-top: 35px;
}
.login-container .inputBox input {
  position: relative;
  width: 100%;
  padding: 20px 10px 10px;
  background: transparent;
  border: none;
  outline: none;
  color: var(--black);
  font-size: 1em;
  letter-spacing: 0.05em;
  z-index: 10;
}
.login-container input[type='submit'] {
  font-size: 20px;
  border: none;
  outline: none;
  background: var(--blue);
  padding: 5px;
  margin-top: 60px;
  border-radius: 90px;
  font-weight: 600;
  cursor: pointer;
}
.login-container input[type='submit']:active {
  background: linear-gradient(90deg, var(--blue), var(--pink));
  opacity: 0.8;
}
.login-container .inputBox span {
  position: absolute;
  left: 0;

```

```

padding: 20px 10px 10px;
font-size: 1em;
color: var(--grey);
pointer-events: none;
letter-spacing: 0.05em;
transition: 0.5s;
}
.login-container .inputBox input:valid ~ span,
.login-container .inputBox input:focus ~ span {
  color: var(--blue);
  transform: translateX(-10px) translateY(-30px);
  font-size: 0.75em;
}
.login-container .inputBox i {
  position: absolute;
  left: 0;
  bottom: 0;
  width: 100%;
  background: var(--blue);
  border-radius: 4px;
  transition: 0.5s;
  pointer-events: none;
  z-index: 9;
}
.login-container .inputBox input:valid ~ i,
.login-container .inputBox input:focus ~ i {
  height: 44px;
}
.login-container .links {
  display: flex;
  justify-content: space-between;
}
.login-container .links a {
  margin: 25px 0;
  font-size: 1em;
  color: var(--grey);
  text-decoration: none;
}
.login-container .links a:hover,
.links a:nth-child(2) {
  color: var(--pink);
}
.register-container {

```



```

    position: relative;
    width: 370px;
    height: 570px;
    background: var(--black);
    border-radius: 50px 5px;
    overflow: hidden;
}
.register-container::before {
    content: "";
    position: absolute;
    top: -50%;
    left: -50%;
    width: 370px;
    height: 570px;
    background: linear-gradient(60deg, transparent, var(--blue), var(--blue));
    transform-origin: bottom right;
    animation: animate 6s linear infinite;
}
.register-container::after {
    content: "";
    position: absolute;
    top: -50%;
    left: -50%;
    width: 370px;
    height: 570px;
    background: linear-gradient(60deg, transparent, var(--pink), var(--pink));
    transform-origin: bottom right;
    animation: animate 6s linear infinite;
    animation-delay: -3s;
}
.register-container form {
    position: absolute;
    inset: 2px;
    border-radius: 50px 5px;
    background: var(--bg);
    z-index: 10;
    padding: 30px 30px;
    display: flex;
    flex-direction: column;
}
.register-container h3 {
    color: var(--blue);
    font-size: 25px;

```

```

    font-weight: 500;
    text-align: center;
}
.register-container .inputBox {
    position: relative;
    width: 300px;
    margin-top: 20px;
}
.register-container .inputBox input {
    position: relative;
    width: 100%;
    padding: 20px 10px 10px;
    background: transparent;
    border: none;
    outline: none;
    color: #23233a;
    font-size: 1em;
    letter-spacing: 0.05em;
    z-index: 10;
}
.register-container input[type='submit'] {
    font-size: 20px;
    border: none;
    outline: none;
    background: var(--blue);
    padding: 5px;
    margin-top: 40px;
    border-radius: 90px;
    font-weight: 600;
    cursor: pointer;
}
.register-container input[type='submit']:active {
    background: linear-gradient(90deg, var(--blue), var(--pink));
    opacity: 0.8;
}
.register-container .inputBox span {
    position: absolute;
    left: 0;
    padding: 20px 10px 10px;
    font-size: 0.8em;
    color: var(--grey);
    pointer-events: none;
    letter-spacing: 0.05em;
}

```

```

    transition: 0.5s;
}
.register-container .inputBox input:valid ~ span,
.register-container .inputBox input:focus ~ span {
    color: var(--blue);
    transform: translateX(-10px) translateY(-30px);
    font-size: 0.75em;
}
.register-container .inputBox i {
    position: absolute;
    left: 0;
    bottom: 0;
    width: 100%;
    background: var(--blue);
    border-radius: 4px;
    transition: 0.5s;
    pointer-events: none;
    z-index: 9;
}
.register-container .inputBox input:valid ~ i,
.register-container .inputBox input:focus ~ i {
    height: 44px;
}
.register-container .links {
    display: flex;
    justify-content: space-between;
}
.register-container .links a {
    margin: 25px 0;
    font-size: 1em;
    color: var(--grey);
    text-decoration: none;
}
.register-container .links a:hover,
.links a:nth-child(2) {
    color: var(--pink);
}
.otp-container {
    position: relative;
    width: 370px;
    height: 450px;
    background: var(--black);
    border-radius: 50px 5px;

```

```

    overflow: hidden;
}
.otp-container::before {
    content: "";
    position: absolute;
    top: -50%;
    left: -50%;
    width: 370px;
    height: 450px;
    background: linear-gradient(60deg, transparent, var(--blue), var(--blue));
    transform-origin: bottom right;
    animation: animate 6s linear infinite;
}
.otp-container::after {
    content: "";
    position: absolute;
    top: -50%;
    left: -50%;
    width: 370px;
    height: 450px;
    background: linear-gradient(60deg, transparent, var(--pink), var(--pink));
    transform-origin: bottom right;
    animation: animate 6s linear infinite;
    animation-delay: -3s;
}
.otp-container form {
    position: absolute;
    inset: 2px;
    border-radius: 50px 5px;
    background: var(--bg);
    z-index: 10;
    padding: 30px 30px;
    display: flex;
    flex-direction: column;
}
.otp-container h3 {
    color: var(--blue);
    font-size: 35px;
    font-weight: 500;
    text-align: center;
}
.otp-container .inputBox {
    position: relative;

```

```

    width: 300px;
    margin-top: 35px;
}
.otp-container .inputBox input {
    position: relative;
    width: 100%;
    padding: 20px 10px 10px;
    background: transparent;
    border: none;
    outline: none;
    color: var(--black);
    font-size: 1em;
    letter-spacing: 0.05em;
    z-index: 10;
}
.otp-container input[type='submit'] {
    font-size: 20px;
    border: none;
    outline: none;
    background: var(--blue);
    padding: 5px;
    margin-top: 40px;
    border-radius: 90px;
    font-weight: 600;
    cursor: pointer;
}
.otp-container input[type='submit']:active {
    background: linear-gradient(90deg, var(--blue), var(--pink));
    opacity: 0.8;
}
.otp-container .inputBox span {
    position: absolute;
    left: 0;
    padding: 20px 10px 10px;
    font-size: 1em;
    color: var(--grey);
    pointer-events: none;
    letter-spacing: 0.05em;
    transition: 0.5s;
}
.otp-container .inputBox input:valid ~ span,
.otp-container .inputBox input:focus ~ span {
    color: var(--blue);
}

```

```

    transform: translateX(-10px) translateY(-30px);
    font-size: 0.75em;
}
.otp-container .inputBox i {
    position: absolute;
    left: 0;
    bottom: 0;
    width: 100%;
    background: var(--blue);
    border-radius: 4px;
    transition: 0.5s;
    pointer-events: none;
    z-index: 9;
}
.otp-container .inputBox input:valid ~ i,
.otp-container .inputBox input:focus ~ i {
    height: 44px;
}
.otp-container .links {
    display: flex;
    justify-content: space-between;
}
.otp-container .links a {
    margin: 25px 0;
    font-size: 1em;
    color: var(--grey);
    text-decoration: none;
}
.otp-container .links a:hover,
.links a:nth-child(2) {
    color: var(--pink);
}
.password-container {
    position: relative;
    width: 370px;
    height: 450px;
    background: var(--black);
    border-radius: 50px 5px;
    overflow: hidden;
}
.password-container::before {
    content: "";
    position: absolute;

```

```

top: -50%;
left: -50%;
width: 370px;
height: 450px;
background: linear-gradient(60deg, transparent, var(--blue), var(--blue));
transform-origin: bottom right;
animation: animate 6s linear infinite;
}
.password-container::after {
  content: "";
  position: absolute;
  top: -50%;
  left: -50%;
  width: 370px;
  height: 450px;
  background: linear-gradient(60deg, transparent, var(--pink), var(--pink));
  transform-origin: bottom right;
  animation: animate 6s linear infinite;
  animation-delay: -3s;
}
.password-container form {
  position: absolute;
  inset: 2px;
  border-radius: 50px 5px;
  background: var(--bg);
  z-index: 10;
  padding: 30px 30px;
  display: flex;
  flex-direction: column;
}
.password-container h3 {
  color: var(--blue);
  font-size: 35px;
  font-weight: 500;
  text-align: center;
}
.password-container .inputBox {
  position: relative;
  width: 300px;
  margin-top: 35px;
}
.password-container .inputBox input {
  position: relative;

```

```

width: 100%;
padding: 20px 10px 10px;
background: transparent;
border: none;
outline: none;
color: var(--black);
font-size: 1em;
letter-spacing: 0.05em;
z-index: 10;
}
.password-container input[type='submit'] {
  font-size: 20px;
  border: none;
  outline: none;
  background: var(--blue);
  padding: 5px;
  margin-top: 75px;
  border-radius: 90px;
  font-weight: 600;
  cursor: pointer;
}
.password-container input[type='submit']:active {
  background: linear-gradient(90deg, var(--blue), var(--pink));
  opacity: 0.8;
}
.password-container .inputBox span {
  position: absolute;
  left: 0;
  padding: 20px 10px 10px;
  font-size: 1em;
  color: var(--grey);
  pointer-events: none;
  letter-spacing: 0.05em;
  transition: 0.5s;
}
.password-container .inputBox input:valid ~ span,
.password-container .inputBox input:focus ~ span {
  color: var(--blue);
  transform: translateX(-10px) translateY(-30px);
  font-size: 0.75em;
}
.password-container .inputBox i {
  position: absolute;

```

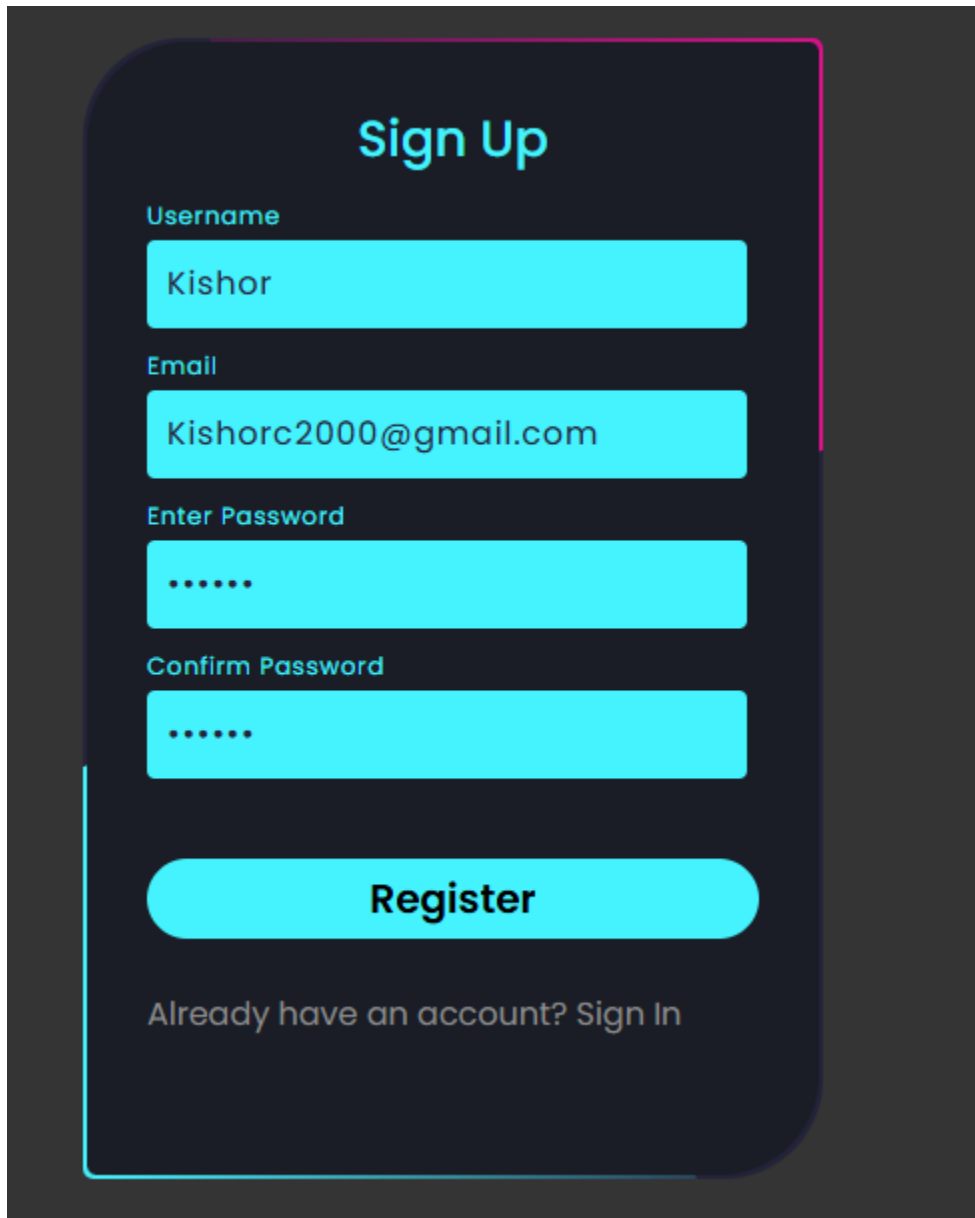


```

    left: 0;
    bottom: 0;
    width: 100%;
    background: var(--blue);
    border-radius: 4px;
    transition: 0.5s;
    pointer-events: none;
    z-index: 9;
}
.password-container .inputBox input:valid ~ i,
.password-container .inputBox input:focus ~ i {
    height: 44px;
}
.password-container .links {
    display: flex;
    justify-content: space-between;
}
.password-container .links a {
    margin: 25px 0;
    font-size: 1em;
    color: var(--grey);
    text-decoration: none;
}
.password-container .links a:hover,
.links a:nth-child(2) {
    color: var(--pink);
}

```

Screenshot



The screenshot shows a registration form titled "Sign Up" on a dark background. The form contains four input fields: "Username" with the value "Kishor", "Email" with the value "Kishorc2000@gmail.com", "Enter Password" with masked characters "*****", and "Confirm Password" with masked characters "*****". Below the fields is a large blue "Register" button. At the bottom, there is a link that says "Already have an account? Sign In".

Fig. Screenshot of registration page

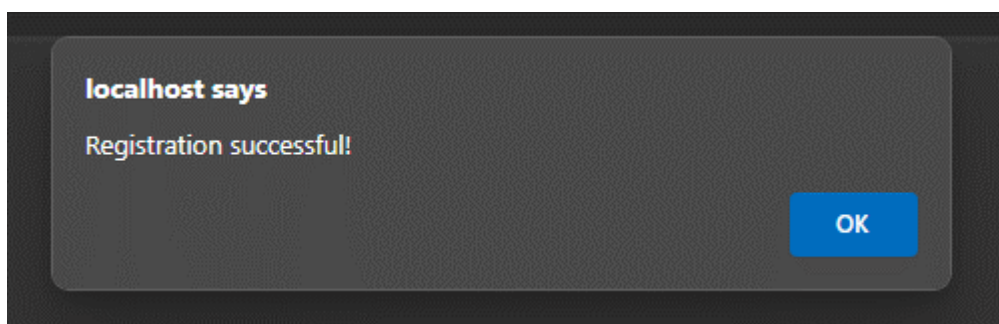


Fig. Screenshot of alert message for registration.

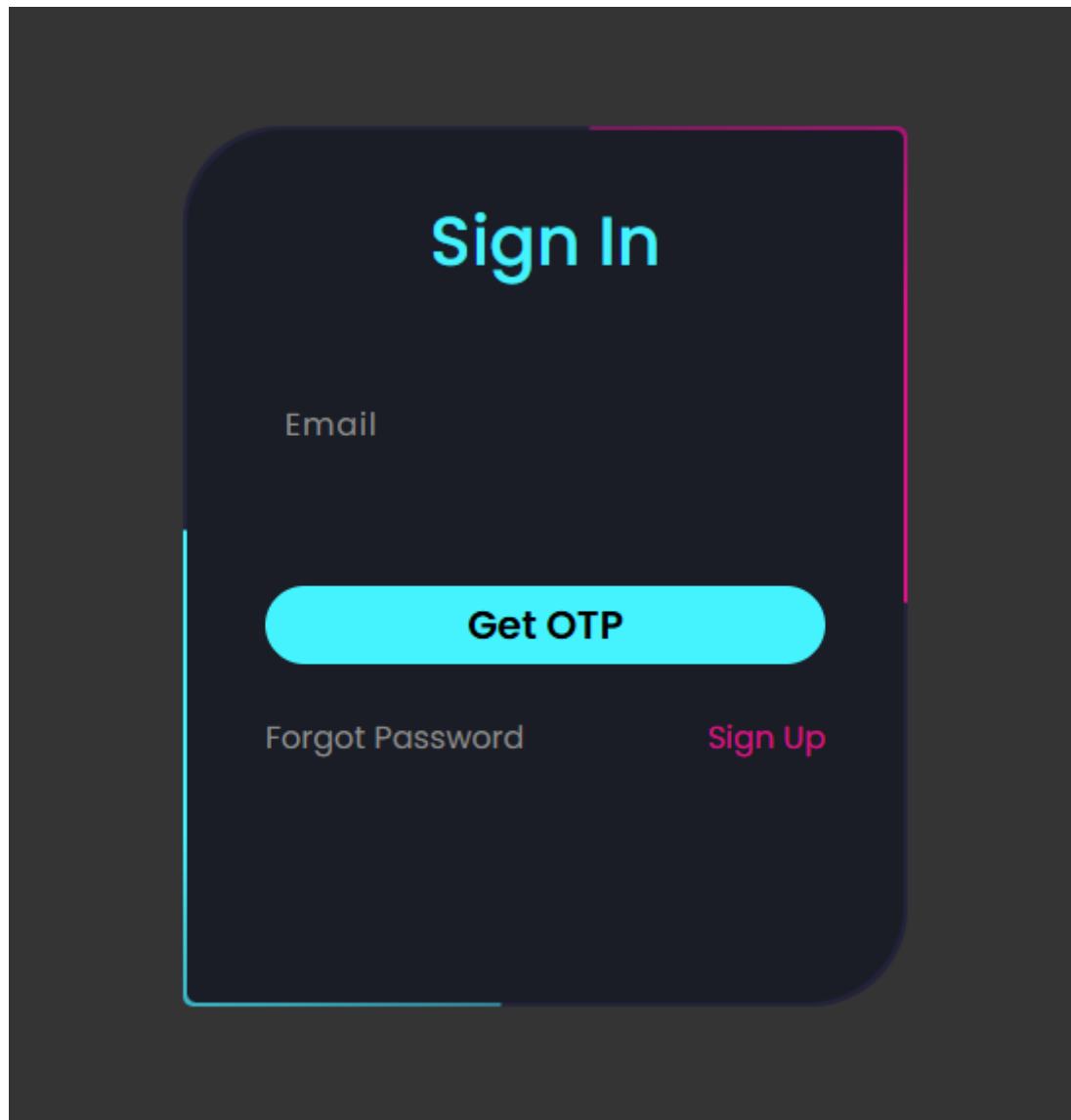


Fig. Login Page

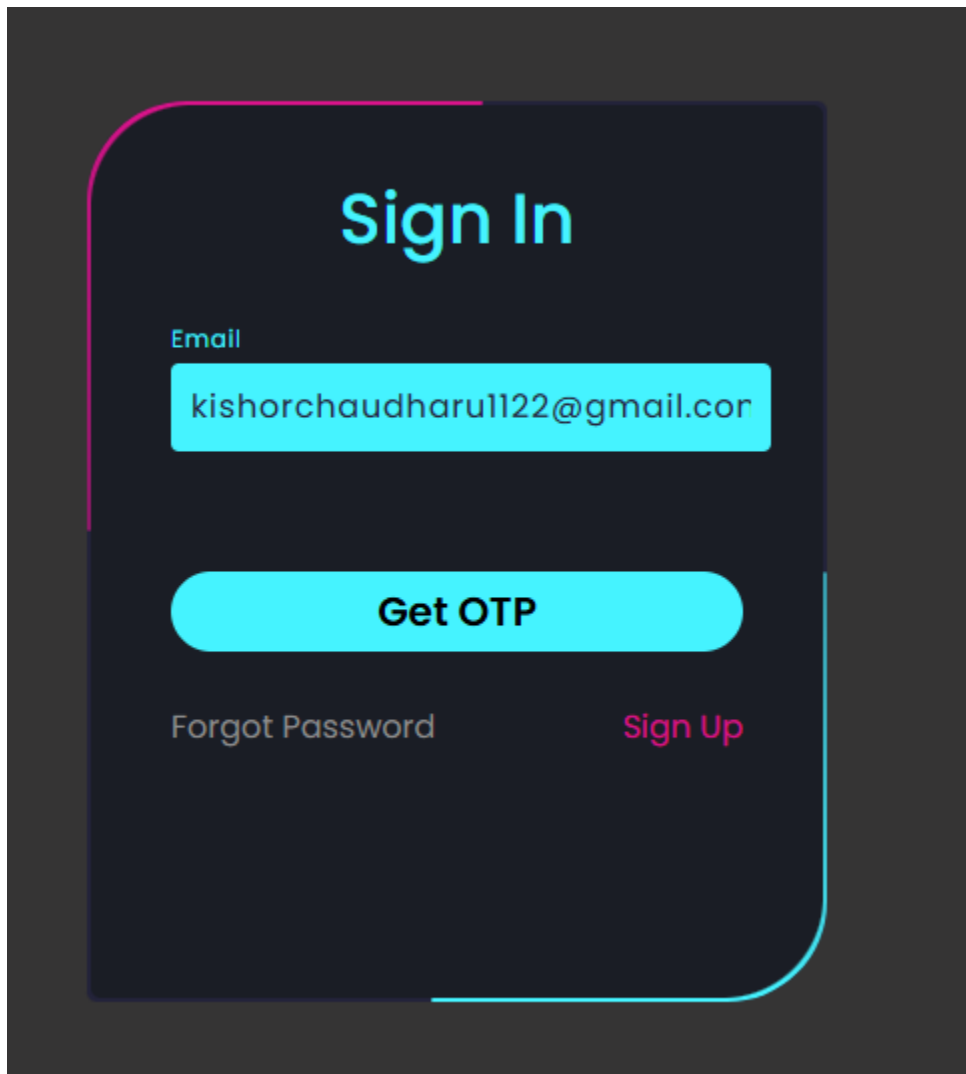


Fig. Screenshot of entering email

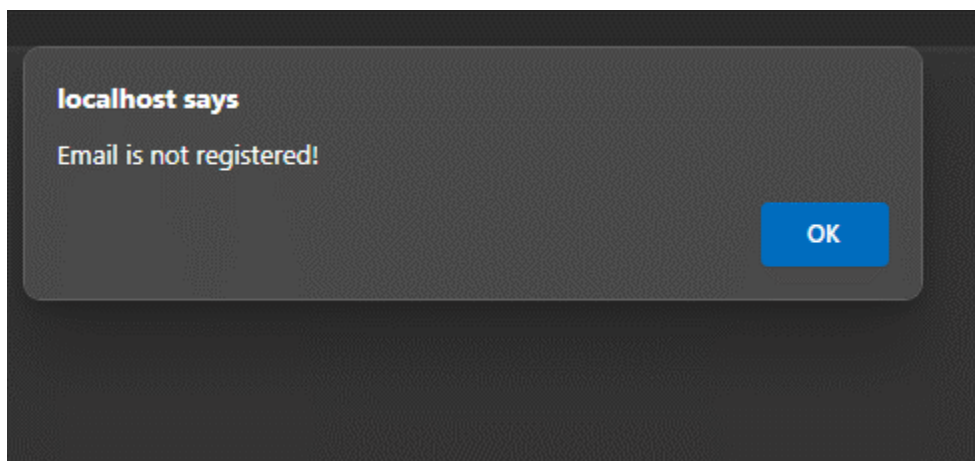


Fig. Screenshot of alert message with not registered email.

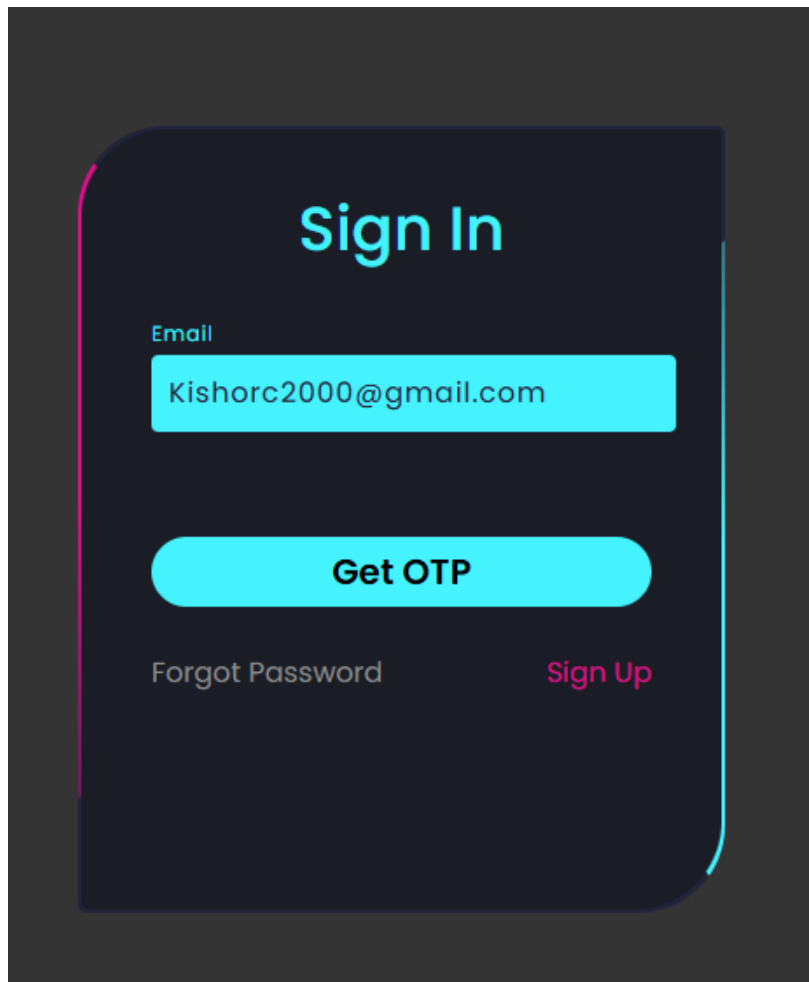


Fig. Screenshot of entering email

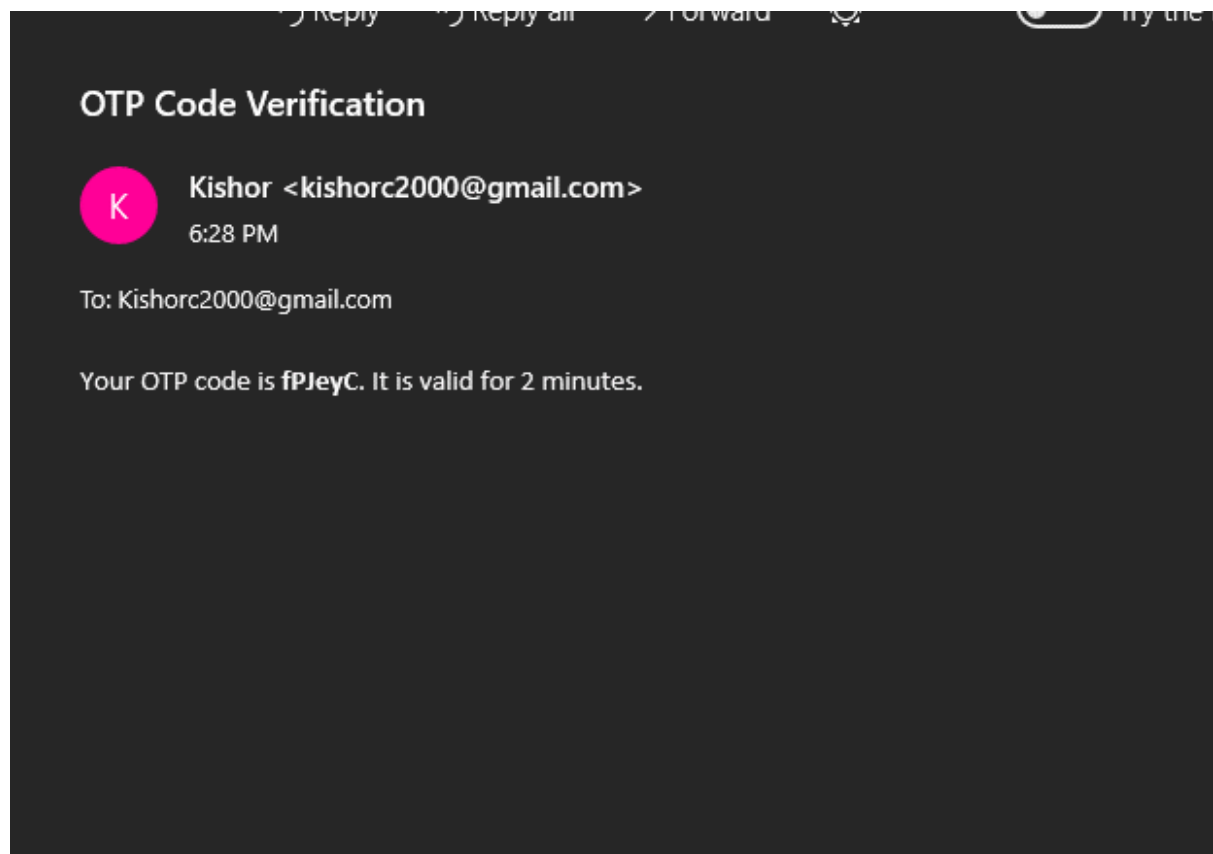


Fig. Screenshot of getting OTP in mail

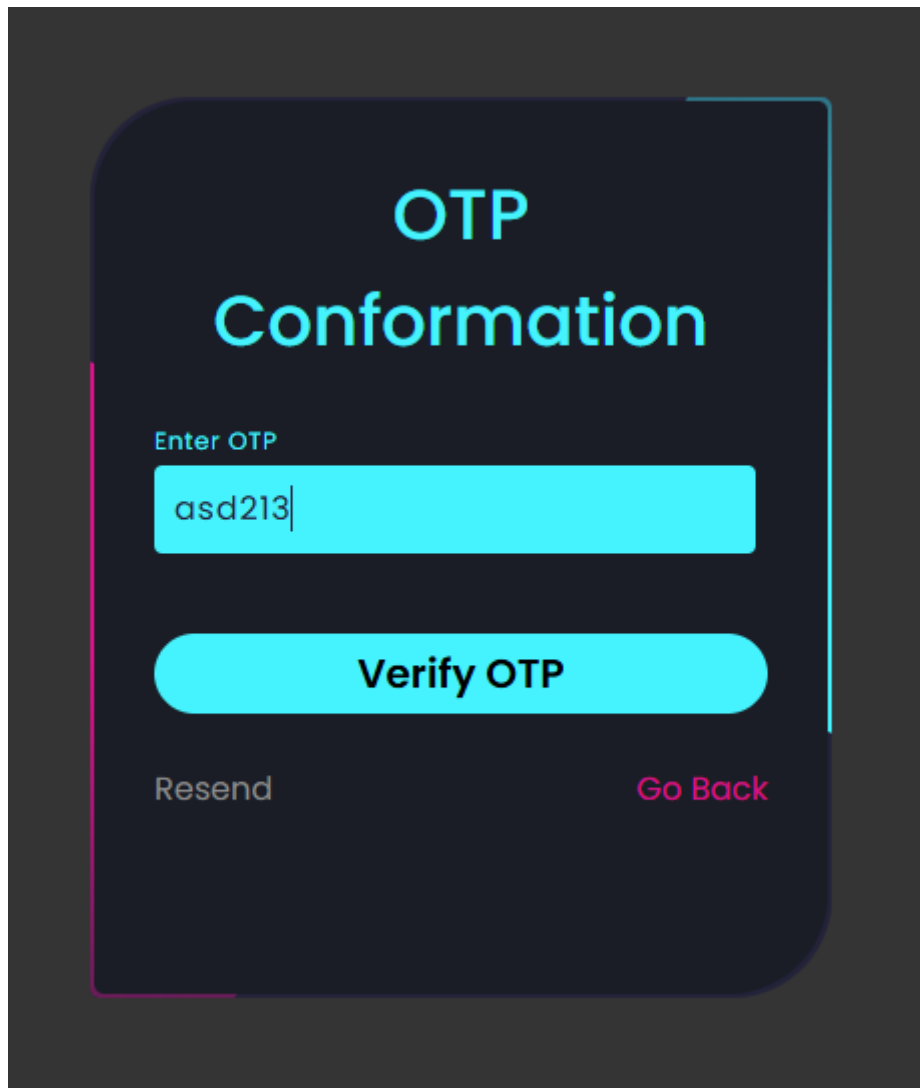


Fig. Screenshot of entering wrong OTP

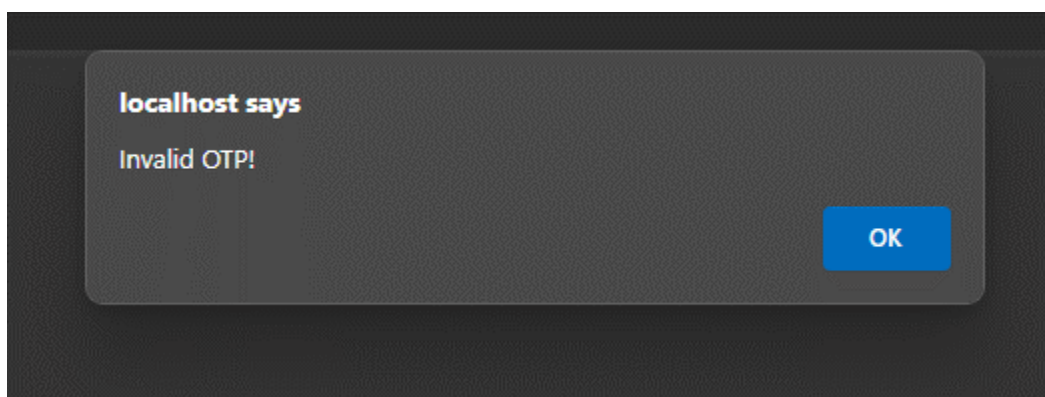


Fig. Screenshot of alert message with wrong OTP

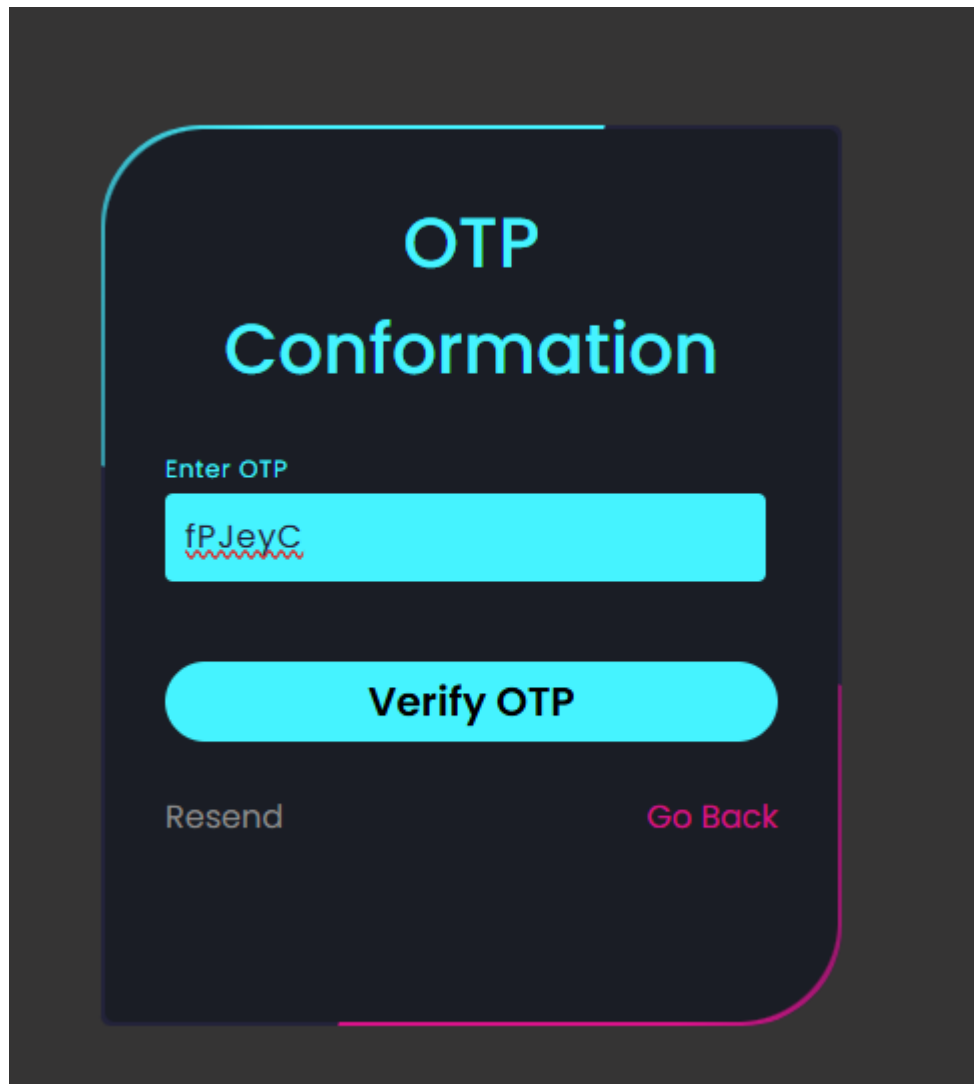


Fig. Screenshot of entering valid OTP

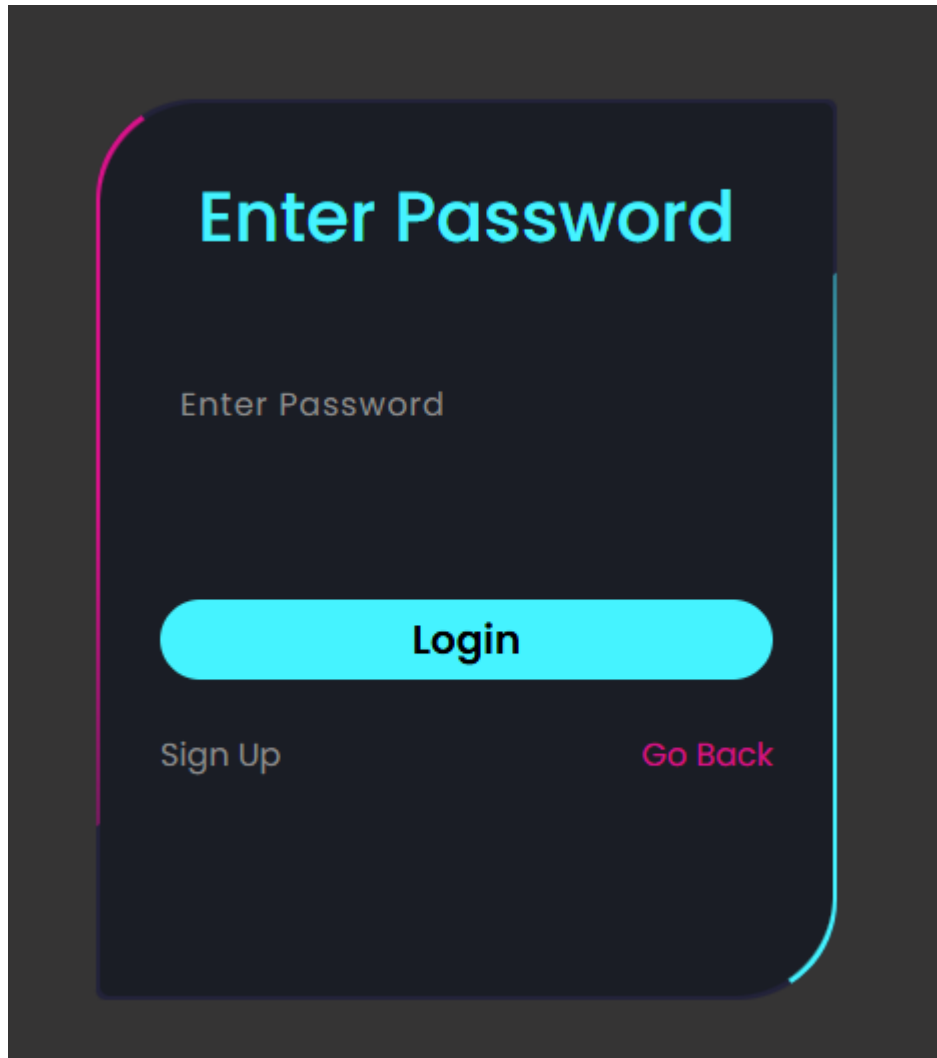


Fig. Screenshot of Password form

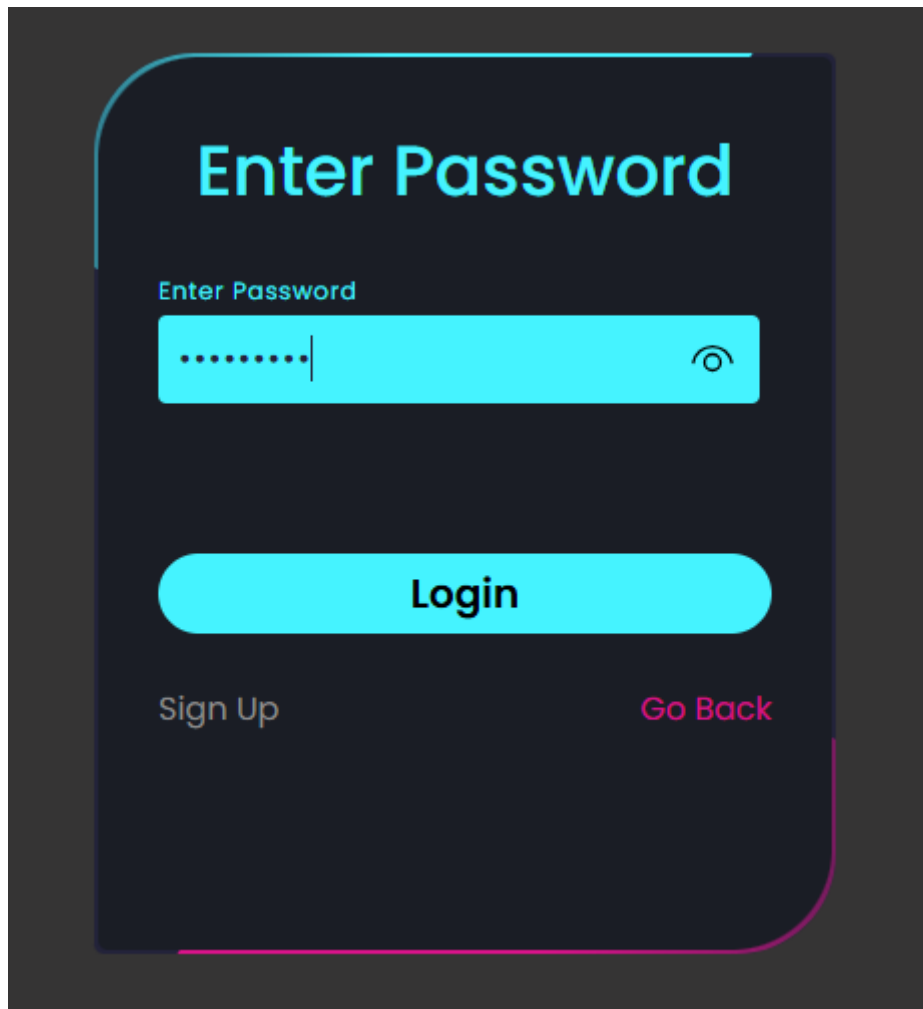


Fig. Screenshot of entering wrong password

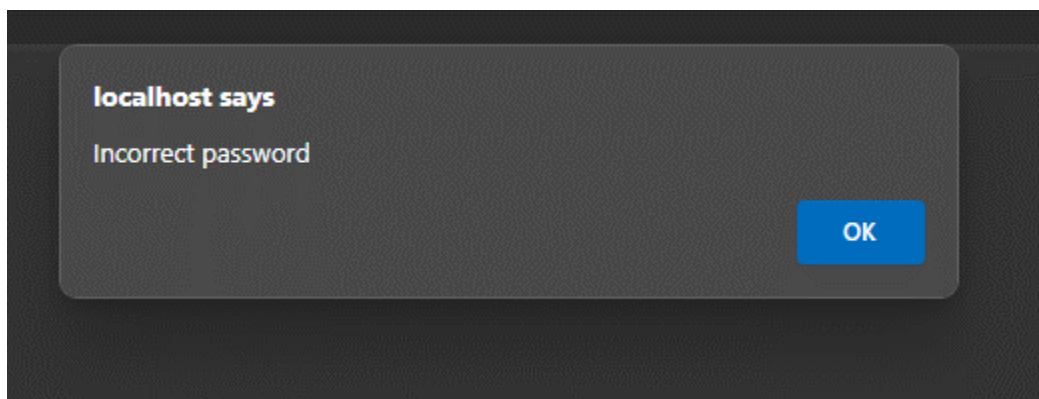


Fig. Screenshot of incorrect password

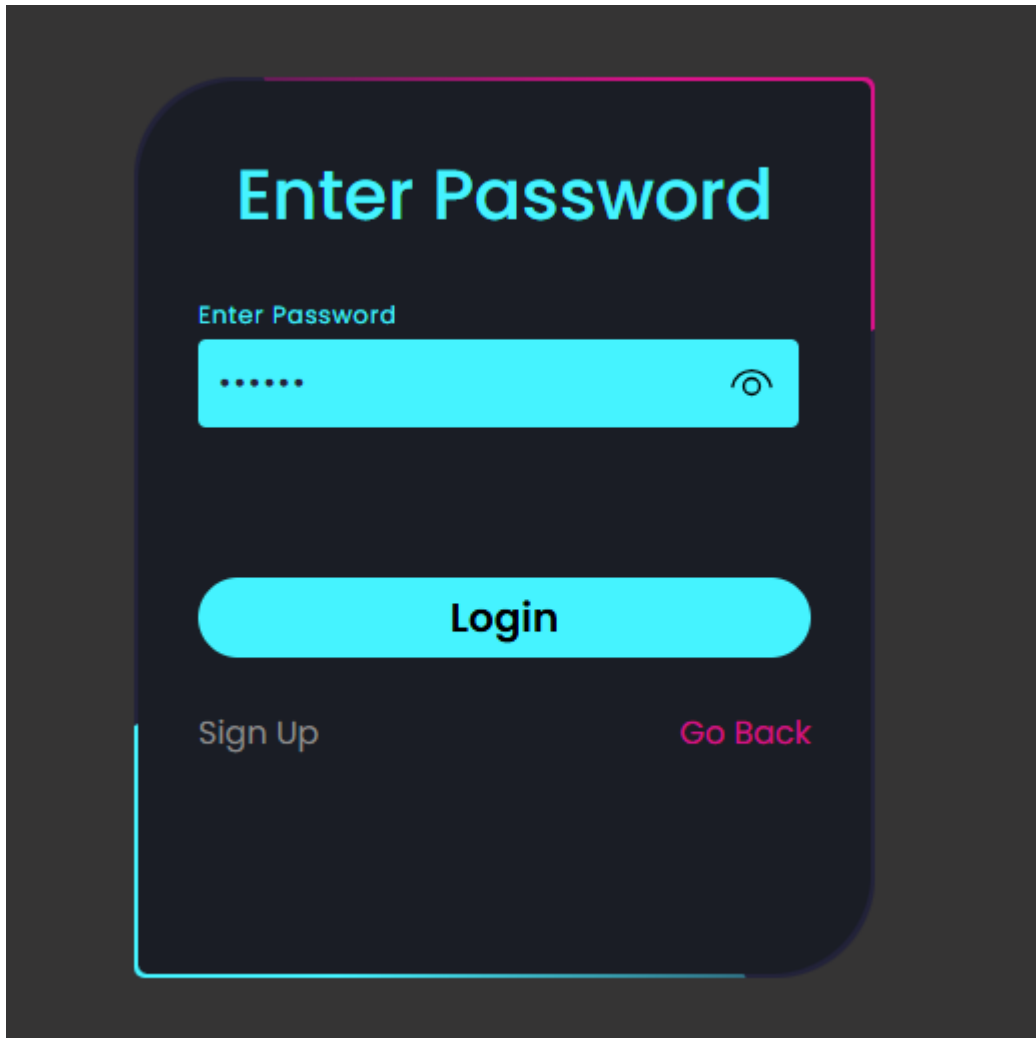


Fig. Screenshot of entering valid password

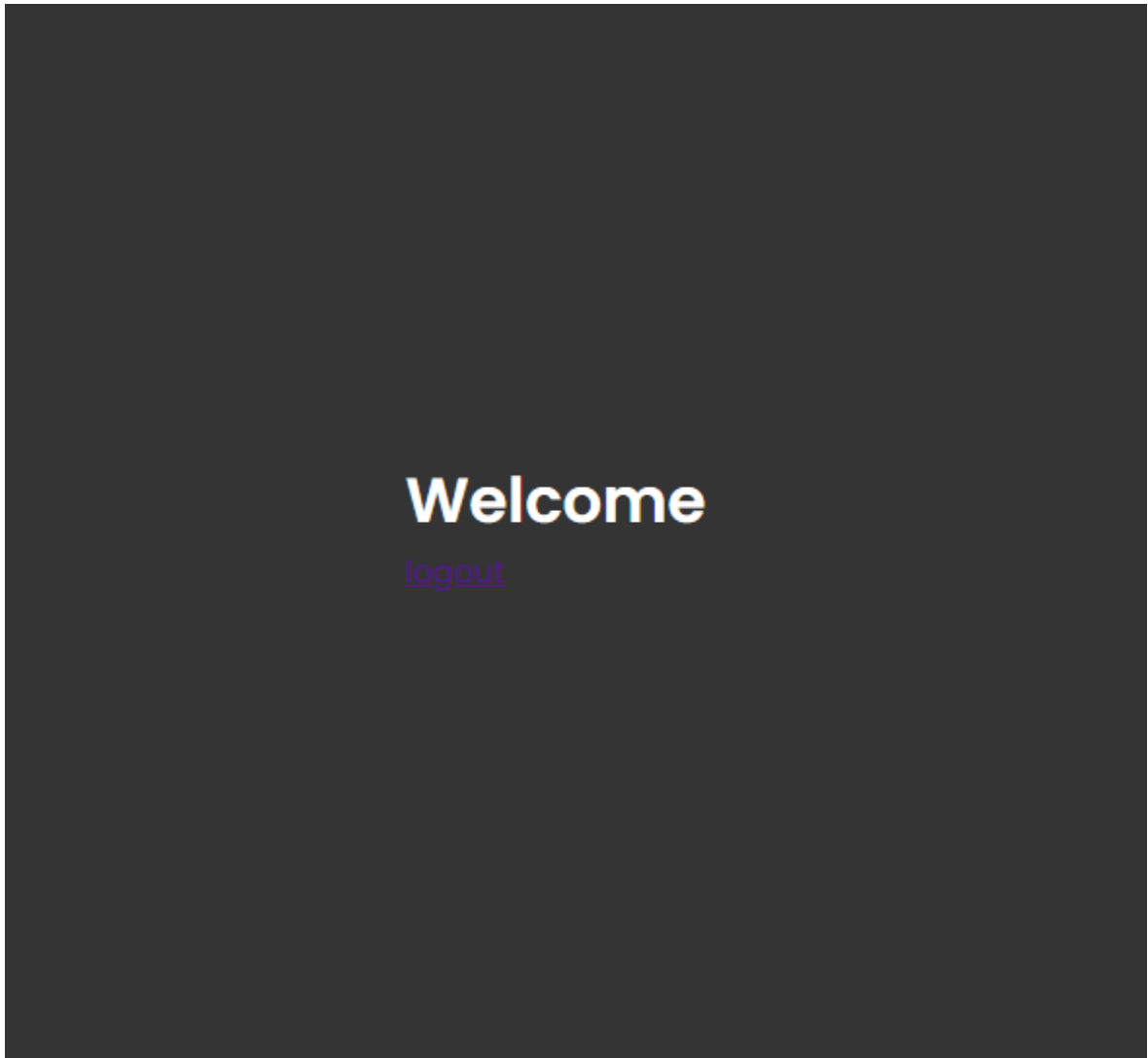


Fig. Screenshot of Homepage

Database Design

test users	test otps
id : int(11)	id : int(11)
name : varchar(255)	email : varchar(100)
email : varchar(255)	otp : varchar(6)
password : varchar(255)	created_at : datetime
created_at : timestamp	

Fig. Screenshot of database design on localhost

Conclusion

Developing a secure and efficient login system crucial part in any web application. With integrating PHP with an OTP verification process enhanced the authentication process. Combination of OTP with password is more secure than other only password system. This will provide a defense against unauthorized access and maintaining system integrity.

This system uses the password hashing, input validation and error handling for a security process which will provide a user security and data protection.

The limitation of this system are OTP delivery reliability, brute force attack, session management, OTP storage, Server load and integration with other system.

Assignment 1 Marking Scheme

Name: Kishor Chaudhary

Student ID: 00020049

Part one: Coding

Assignment Criteria	Maximum Marks	Student Mark
Data validation	5	
Navigation	5	
Database design	8	
OTP Generation	5	
OTP validation	7	
Accuracy	6	
Execution	10	
Interface design	12	
Error handling for invalid input	6	
Code structure and organization	6	
Total	70	

Part Two: Documentation

Assignment Criteria	Maximum Marks	Student Mark
Cover page	1	
Table of contents	1	
Introduction	4	
Objective	3	
Workflow	3	
Coding	7	
Conclusion	5	
References	1	
Images	3	
Formatting	2	
Total	30	