

C. Solution/Code:

maincontent.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Secure File Management</title>
    <link rel="stylesheet" href="styles.css">
</head>
<body>
    <div class="container">
        <h2>Secure File Management System</h2>

        <!-- Authentication Section -->
        <div id="auth-section">
            <div id="signup">
                <h3>Sign Up</h3>
                <input type="text" id="newUsername" placeholder="Create Username">
                <input type="password" id="newPassword" placeholder="Create Password">
                <button onclick="signUp()">Sign Up</button>
                <p>Already have an account? <a href="#" onclick="showLogin()">Login</a></p>
            </div>

            <div id="login" class="hidden">
                <h3>Login</h3>
                <input type="text" id="username" placeholder="Enter Username">
                <input type="password" id="password" placeholder="Enter Password">
                <button onclick="authenticate()">Login</button>
                <p>Don't have an account? <a href="#" onclick="showSignup()">Sign Up</a></p>
            </div>
        </div>

        <!-- File Management Section -->
        <div id="file-section" class="hidden">
            <h3>Welcome, <span id="userDisplay"></span></h3>
            <button id="logout" onclick="logout()">Logout</button>
            <input type="file" id="fileInput" multiple>
            <button onclick="encryptFiles()">Encrypt & Save</button>
            <p id="status"></p>
            <div id="fileList"></div>
        </div>
    </div>

    <script src="https://cdnjs.cloudflare.com/ajax/libs/crypto-js/4.1.1/crypto-js.min.js"></script>
    <script src="script.js"></script>
</body>
</html>
```

Index1.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Secure File Management</title>
    <link rel="stylesheet" href="css/slideshow.css">
</head>
<body>

    <!-- Navbar -->
    <nav class="navbar">
        <div class="logo">Secure File Management System</div>
        <div class="nav-links">
            <a href="login.php" style="border-style:solid; padding:5px; border-width:0.5px; border-radius:5px;">Login</a>
            <a href="signup.php" style="border-style:solid; padding:5px; border-width:0.5px; border-radius:5px;">Sign up</a>
        </div>
    </nav>

    <!-- Slideshow Container -->
    <div class="slideshow-container">

        <!-- Slide 1: Intro Section -->
        <div class="slide fade">
```

```

<div class="intro">
  <h1>Secure & Manage Your Files</h1>
  <p>End-to-end encrypted cloud storage with advanced security.</p>
  <button class="btn" onclick="window.location.href='login.php';">Get Started</button>
</div>
</div>

<!-- Slide 2: Login Section -->
<div class="slide fade">
  <div class="login-slide">
    <h1> <span style="font-size: 80px;"> 📁 </span> Log in & Secure Your Files</h1>
    <br>
    <p>Join us and store your files safely with encryption.</p>
    <br>
    <button class="btn" onclick="window.location.href='login.php';">Login Now</button>
  </div>
</div>

<!-- Slide 3: Features Section -->
<div class="slide fade">
  <div class="features">
    <h2>Why Choose Us?</h2>
    <div class="feature-box">
      <span> 🔒 </span> <p>Advanced Encryption</p>
    </div>
    <div class="feature-box">
      <span> 🌐 </span> <p>Malware Protection</p>
    </div>
    <div class="feature-box">
      <span> 2FA </span> <p>2-Factor Authentication</p>
    </div>
  </div>
</div>

<!-- Navigation Arrows -->
<a class="prev" onclick="changeSlide(-1)">#10094;</a>
<a class="next" onclick="changeSlide(1)">#10095;</a>

</div>

<!-- Footer -->
<footer>
  <p>© 2025 Secure File Management. All rights reserved.</p>
</footer>

<script src="script/slideshow.js"></script>

```

</body>
</html>

styles.css

```

body {
  font-family: 'Poppins', sans-serif;
  background-color: #f4f4f4;
  display: flex;
  justify-content: center;
  align-items: center;
  height: 100vh;
}

.container {
  width: 400px;
  background: white;
  padding: 20px;
  border-radius: 10px;
  box-shadow: 0 4px 10px rgba(0, 0, 0, 0.1);
  text-align: center;
}

h2 {
  margin-bottom: 20px;
}

input, button {
  width: 90%;
  margin: 10px 0;
  padding: 10px;
  border-radius: 5px;
  border: 1px solid #ccc;
}

button {
  cursor: pointer;
  transition: 0.3s;
  background-color: #007BFF;
}

```

```
        color: white;
    }

    button:hover {
        background-color: #0056b3;
    }

    #logout {
        background-color: #dc3545;
    }

    #logout:hover {
        background-color: #b02a37;
    }

    .file-table {
        width: 100%;
        margin-top: 20px;
        border-collapse: collapse;
    }

    .file-table th, .file-table td {
        padding: 10px;
        border-bottom: 1px solid #ddd;
        text-align: left;
    }

    .actions {
        display: flex;
        gap: 10px;
    }

    .download-btn {
        background-color: #28a745;
        color: white;
        padding: 5px 10px;
        border-radius: 5px;
    }

    .delete-btn {
        background-color: #dc3545;
        color: white;
        padding: 5px 10px;
        border-radius: 5px;
    }

    .hidden {
        display: none;
    }

    .share-btn {
        background-color: #ffc107;
        color: black;
        padding: 5px 10px;
        border-radius: 5px;
    }

    .share-btn:hover {
        background-color: #e0a800;
    }

    .share-options {
        margin-top: 10px;
        display: flex;
        gap: 10px;
    }

    .share-icon {
        text-decoration: none;
        padding: 5px 10px;
        border-radius: 5px;
        color: white;
        font-size: 14px;
    }

    .whatsapp {
        background-color: #25D366;
    }

    .facebook {
        background-color: #1877F2;
    }

    .email {
        background-color: #D44638;
    }

    .share-icon:hover {
```

```

    opacity: 0.8;
}

/* Add these to your existing styles.css */
.share-popup {
  background: white;
  border: 1px solid #ddd;
  border-radius: 8px;
  padding: 15px;
  margin-top: 15px;
  box-shadow: 0 2px 10px rgba(0,0,0,0.1);
}

.share-popup h4 {
  margin-top: 0;
  color: #333;
}

.share-buttons {
  display: flex;
  gap: 10px;
  margin: 15px 0;
  flex-wrap: wrap;
}

.share-note {
  font-size: 12px;
  color: #666;
  margin-bottom: 0;
}

.copy-btn {
  background-color: #6c757d;
  color: white;
  border: none;
  padding: 5px 10px;
  border-radius: 5px;
  cursor: pointer;
}

.copy-btn:hover {
  background-color: #5a6268;
}

#file-info {
  text-align: center;
  margin-top: 20px;
}

#error-message {
  color: #dc3545;
  margin-top: 10px;
}

```

script.js

```

let users = JSON.parse(localStorage.getItem("users")) || {};
let loggedinUser = "";
let storedFiles = JSON.parse(localStorage.getItem("storedFiles")) || {};

function showSignup() {
  document.getElementById("signup").classList.remove("hidden");
  document.getElementById("login").classList.add("hidden");
}

function showLogin() {
  document.getElementById("signup").classList.add("hidden");
  document.getElementById("login").classList.remove("hidden");
}

function signUp() {
  let newUsername = document.getElementById("newUsername").value.trim();
  let newPassword = document.getElementById("newPassword").value;

  if (!newUsername || !newPassword) {
    alert("Username and password cannot be empty.");
    return;
  }

  if (users[newUsername]) {
    alert("Username already exists!");
  } else {
    users[newUsername] = newPassword;
    localStorage.setItem("users", JSON.stringify(users));
    alert("Account created successfully! Please log in.");
    showLogin();
  }
}

```

```

function authenticate() {
    let username = document.getElementById("username").value.trim();
    let password = document.getElementById("password").value;

    if (users[username] && users[username] === password) {
        loggedInUser = username;
        document.getElementById("auth-section").classList.add("hidden");
        document.getElementById("file-section").classList.remove("hidden");
        document.getElementById("userDisplay").innerText = loggedInUser;
        loadStoredFiles();
    } else {
        alert("Incorrect username or password!");
    }
}

function logout() {
    loggedInUser = "";
    document.getElementById("file-section").classList.add("hidden");
    document.getElementById("auth-section").classList.remove("hidden");
    document.getElementById("fileList").innerHTML = "";
    showLogin();
}

function encryptFiles() {
    const files = document.getElementById("fileInput").files;
    if (files.length === 0) {
        alert("Select at least one file");
        return;
    }

    if (!storedFiles[loggedInUser]) storedFiles[loggedInUser] = [];

    Array.from(files).forEach(file => {
        const reader = new FileReader();
        reader.onload = function(event) {
            const base64String = btoa(event.target.result);
            const encrypted = CryptoJS.AES.encrypt(base64String, users[loggedInUser]).toString();
            const timestamp = new Date().toLocaleString();
            storedFiles[loggedInUser].push({ name: file.name, data: encrypted, type: file.type, timestamp });
            saveStoredFiles();
            loadStoredFiles();
        };
        reader.readAsBinaryString(file);
    });
}

function decryptFile(index) {
    const file = storedFiles[loggedInUser][index];
    const decrypted = CryptoJS.AES.decrypt(file.data, users[loggedInUser]).toString(CryptoJS.enc.Utf8);
    const binaryData = atob(decrypted);
    const arrayBuffer = new Uint8Array(binaryData.length);
    for (let i = 0; i < binaryData.length; i++) {
        arrayBuffer[i] = binaryData.charCodeAt(i);
    }
    const blob = new Blob([arrayBuffer], { type: file.type });
    const a = document.createElement("a");
    a.href = URL.createObjectURL(blob);
    a.download = file.name;
    a.click();
}

function deleteFile(index) {
    storedFiles[loggedInUser].splice(index, 1);
    saveStoredFiles();
    loadStoredFiles();
}

function shareFile(index) {
    const file = storedFiles[loggedInUser][index];

    if (!file) {
        alert("File not found!");
        return;
    }

    const encryptedData = encodeURIComponent(file.data);
    const fileName = encodeURIComponent(file.name);
    const shareableLink = `${window.location.origin}/share.html?file=${fileName}&data=${encryptedData}`;

    const whatsappLink = `https://wa.me/?text=Download%20file%20${fileName}%20from%20${shareableLink}`;
    const facebookLink = `https://www.facebook.com/sharer/sharer.php?u=${shareableLink}`;
    const gmailUrl = `https://mail.google.com/mail/?view=cm&fs=1&tf=1&su=Shared%20File%20${file.name}&body=Download%20the%20file%20from%20the%20link%20below%20${shareableLink}`;

    const shareOptions = `
        <div class="share-options">
            <p>Share via:</p>
            <a href="${whatsappLink}" target="_blank" class="share-icon whatsapp">WhatsApp</a>
    `;
}

```

```

<a href="${facebookLink}" target="_blank" class="share-icon facebook">Facebook</a>
<a href="${gmailUrl}" target="_blank" class="share-icon email">Email</a>
</div>';

document.getElementById("fileList").insertAdjacentHTML("beforeend", shareOptions);
}

function saveStoredFiles() {
  localStorage.setItem("storedFiles", JSON.stringify(storedFiles));
}

function loadStoredFiles() {
  storedFiles = JSON.parse(localStorage.getItem("storedFiles")) || {};
  let fileTable =
    `<table class="file-table">
      <tr>
        <th>File Name</th>
        <th>Timestamp</th>
        <th>Actions</th>
      </tr>`;
  if (storedFiles[loggedInUser]) {
    storedFiles[loggedInUser].forEach((file, index) => {
      fileTable += `
        <tr>
          <td>${file.name}</td>
          <td>${file.timestamp}</td>
          <td class="actions">
            <button class="download-btn" onclick="decryptFile(${index})">⬇</button>
            <button class="delete-btn" onclick="deleteFile(${index})">✖</button>
            <button class="share-btn" onclick="shareFile(${index})">🔗</button>
          </td>
        </tr>`;
    });
  }
  fileTable += `</table>`;
  document.getElementById("fileList").innerHTML = fileTable;
}

```

signup.php

```

<?php
include 'db_config.php';

if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $email = $_POST['email'];
  $generated_password = $_POST['generated_password'];
  $created_password = $_POST['created_password'];
  $confirm_password = $_POST['confirm_password'];

  if (!empty($email) && !empty($generated_password) && !empty($created_password) && !empty($confirm_password)) {
    // Check if email already exists
    $check_sql = "SELECT id FROM users WHERE email = ?";
    $check Stmt = $conn->prepare($check_sql);
    $check Stmt->bind_param("s", $email);
    $check Stmt->execute();
    $check Stmt->store_result();

    if ($check Stmt->num_rows > 0) {
      echo "<script>alert('This email is already registered! Please login.');" . window.location.href = 'login.php' . "</script>";
      exit();
    }
    $check Stmt->close();
  }

  if ($created_password === $confirm_password) {
    // Hash the password
    $hashed_password = password_hash($created_password, PASSWORD_DEFAULT);

    // Insert user into database
    $sql = "INSERT INTO users (email, password) VALUES (?, ?)";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("ss", $email, $hashed_password);

    if ($stmt->execute()) {
      echo "<script>";
      alert('Sign up successful! Redirecting to login..');
      window.location.href = 'login.php';
      "</script>";
      exit();
    } else {
      echo "<script>alert('Error: Could not register. Try again!');</script>";
    }
    $stmt->close();
  } else {
    echo "<script>alert('Passwords do not match!');</script>";
  }
} else {

```

```

echo "<script>alert('Please fill all fields correctly!');</script>";
}
$conn->close();
?>

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Sign Up</title>
<link rel="stylesheet" href="styles.css"> <!-- External CSS -->
<style>
/* General Styles */
* {
margin: 0;
padding: 0;
box-sizing: border-box;
font-family: 'Poppins', sans-serif;
}
body {
display: flex;
align-items: center;
justify-content: center;
height: 100vh;
background: linear-gradient(135deg, #d6e0ea, #87d7f7);
}
.signup-container {
background: #fff;
padding: 30px;
border-radius: 10px;
box-shadow: 0 5px 15px rgba(0, 0, 0, 0.2);
width: 400px;
text-align: center;
}
.signup-container h2 {
margin-bottom: 15px;
color: #333;
}
.signup-container input {
width: 100%;
padding: 12px;
margin: 10px 0;
border: 1px solid #ddd;
border-radius: 5px;
font-size: 16px;
}
.signup-container input:focus {
border-color: #007bff;
outline: none;
}
.signup-container button {
width: 100%;
padding: 12px;
background: #007bff;
border: none;
color: #fff;
font-size: 18px;
border-radius: 5px;
cursor: pointer;
transition: 0.3s;
}
.signup-container button:hover {
background: #0056b3;
}
.redirect-link {
display: block;
margin-top: 15px;
color: #007bff;
text-decoration: none;
font-size: 14px;
}
.redirect-link:hover {
text-decoration: underline;
}
</style>
</head>
<body>

<div class="signup-container">
<h2>Sign Up</h2>
<form action="signup.php" method="POST" onsubmit="return validatePasswords()">
<input type="email" name="email" placeholder="Enter your email" required>

<!-- Auto-generated password (User can change it) -->
<input type="text" name="generated_password" id="generated_password" placeholder="Generated Password" readonly>
<button type="button" onclick="generatePassword()">Generate Password</button>

```

```

<input type="password" name="created_password" id="created_password" placeholder="Create Password" required>
<input type="password" name="confirm_password" id="confirm_password" placeholder="Confirm Password" required>
<button type="submit">Sign Up</button>
</form>
<a href="login.php" class="redirect-link">Already have an account? Login</a>
</div>

<script>
function generatePassword() {
    let chars = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@#$_";
    let password = "";
    for (let i = 0; i < 10; i++) {
        password += chars.charAt(Math.floor(Math.random() * chars.length));
    }
    document.getElementById("generated_password").value = password;
}

function validatePasswords() {
    let pass1 = document.getElementById("created_password").value;
    let pass2 = document.getElementById("confirm_password").value;

    if (pass1 !== pass2) {
        alert("Passwords do not match!");
        return false;
    }
    return true;
}
</script>

</body>
</html>

```

login.php

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login</title>

    <link rel="stylesheet" href="css/login.css">
</head>
<body>

    <div class="login-container">
        <h2>Login</h2>
        <form action="send_otp.php" method="POST">
            <input type="email" name="email" placeholder="Enter your email" required>
            <input type="password" name="password" placeholder="Enter your password" required>
            <button type="submit">Login</button>
        </form>
        <a href="#">Forgot Password?</a>
    </div>

</body>
</html>

```

sent_otp.php

```

<?php
session_start();
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\Exception;

require 'vendor/autoload.php'; // PHPMailer include

include 'db_config.php'; // Database connection include

if (isset($_POST['email'])) {
    $email = $_POST['email'];

    // Check if user exists in database
    $sql = "SELECT id FROM users WHERE email = ?";
    $stmt = $conn->prepare($sql);
    $stmt->bind_param("s", $email);
    $stmt->execute();
    $stmt->store_result();

    if ($stmt->num_rows > 0) {
        // User exists, proceed with OTP
        $otp = rand(100000, 999999); // Generate 6-digit OTP
        $_SESSION['otp'] = $otp;
        $_SESSION['email'] = $email;

        $mail = new PHPMailer(true);

```

```

try {
    // SMTP Configuration
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = 'maxrushto77@gmail.com'; // Your email
    $mail->Password = 'pfvflysxkqkeczad'; // Use App Password
    $mail->SMTPSecure = 'tls';
    $mail->Port = 587;

    // Email Settings
    $mail->setFrom('your-email@gmail.com', 'Your Name');
    $mail->addAddress($email);
    $mail->Subject = 'Your OTP Code';
    $mail->Body = "Your OTP code is: $otp";

    // Send Email
    $mail->send();
    header("Location: verify.php");
    exit();
} catch (Exception $e) {
    echo "<script>alert('Error sending OTP: {$mail->ErrorInfo}');</script>";
}
} else {
    // User does not exist, redirect to signup
    echo "<script>
        alert('No account found! Please sign up.');
        window.location.href = 'signup.php';
    </script>";
}
$stmt->close();
$conn->close();
}
?>

```

verify_otp.php

```

<?php
session_start();
if (isset($_POST['otp'])) {
    $entered_otp = $_POST['otp'];
    if ($entered_otp == $_SESSION['otp']) {
        header("Location: main_content.html");
    } else {
        echo "Invalid OTP! Try again.";
    }
}
?>

```

verify.php

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Verify OTP</title>
    <style>
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
            font-family: 'Poppins', sans-serif;
        }
        body {
            display: flex;
            align-items: center;
            justify-content: center;
            height: 100vh;
            background: linear-gradient(135deg, #d6e0ea, #87d7f7);
        }
        .otp-container {
            background: #fff;
            padding: 30px;
            border-radius: 10px;
            box-shadow: 0 5px 15px rgba(0, 0, 0, 0.2);
            width: 350px;
            text-align: center;
        }
        .otp-container h2 {
            margin-bottom: 15px;
            color: #333;
        }
        .otp-container input {
            width: 100%;
            padding: 12px;
        }
    </style>

```

```

margin: 10px 0;
border: 1px solid #ddd;
border-radius: 5px;
font-size: 16px;
text-align: center;
letter-spacing: 3px;
font-weight: bold;
}
.otp-container input:focus {
border-color: #007bff;
outline: none;
}
.otp-container button {
width: 100%;
padding: 12px;
background: #007bff;
border: none;
color: #fff;
font-size: 18px;
border-radius: 5px;
cursor: pointer;
transition: 0.3s;
}
.otp-container button:hover {
background: #0056b3;
}
.resend-link {
display: block;
margin-top: 15px;
color: #007bff;
text-decoration: none;
font-size: 14px;
}
.resend-link:hover {
text-decoration: underline;
}

```

</style>

</head>

<body>

```

<div class="otp-container">
<h2>Verify OTP</h2>
<p>Please enter the OTP sent to your email.</p>
<form action="verify_otp.php" method="POST">
<input type="text" name="otp" placeholder="Enter OTP" required maxlength="6">
<button type="submit">Verify OTP</button>
</form>
<a href="#" class="resend-link">Resend OTP</a>
</div>

```

</body>

</html>

db_config.php

```

<?php
$host = "localhost";
$user = "root";
$pass = "";
$dbname = "user_database";

$conn = new mysqli($host,$user,$pass,$dbname);

if($conn->connect_error){
die("connection failed: ".$conn->connect_error);
}

```

composer.lock

```

{
"_readme": [
"This file locks the dependencies of your project to a known state",
"Read more about it at https://getcomposer.org/doc/01-basic-usage.md#installing-dependencies",
"This file is @generated automatically"
],
"content-hash": "baf0b1b659e688a64051c5ee9742a77e",
"packages": [
{
"name": "phpmailer/phpmailer",
"version": "v6.9.3",
"source": {
"type": "git",
"url": "https://github.com/PHPMailer/PHPMailer.git",
"reference": "2f5c94fe7493efc213f643c23b1b1c249d40f47e"
},
"dist": {
"type": "zip",

```

```

"url": "https://api.github.com/repos/PHPMailer/PHPMailer/zipball/2f5c94fe7493efc213f643c23b1b1c249d40f47e",
"reference": "2f5c94fe7493efc213f643c23b1b1c249d40f47e",
"shasum": "",
},
"require": {
  "ext-ctype": "*",
  "ext-filter": "*",
  "ext-hash": "*",
  "php": ">=5.5.0"
},
"require-dev": {
  "dealerdirect/phpcodesniffer-composer-installer": "^1.0",
  "doctrine/annotations": "^1.2.6 || ^1.13.3",
  "php-parallel-lint/php-console-highlighter": "^1.0.0",
  "php-parallel-lint/php-parallel-lint": "1.3.2",
  "phpcompatibility/php-compatibility": "^9.3.5",
  "roave/security-advisories": "dev-latest",
  "squizlabs/php_codesniffer": "^3.7.2",
  "yosayt/phunit-polyfills": "1.0.4"
},
"suggest": {
  "decomplexity/SendOauth2": "Adapter for using XOAUTH2 authentication",
  "ext-mbstring": "Needed to send email in multibyte encoding charset or decode encoded addresses",
  "ext-openssl": "Needed for secure SMTP sending and DKIM signing",
  "grew/oauth2-azure-provider": "Needed for Microsoft Azure XOAUTH2 authentication",
  "hayageek/oauth2-yahoo": "Needed for Yahoo XOAUTH2 authentication",
  "league/oauth2-google": "Needed for Google XOAUTH2 authentication",
  "psr/log": "For optional PSR-3 debug logging",
  "symfony/polyfill-mbstring": "To support UTF-8 if the Mbstring PHP extension is not enabled (^1.2)",
  "thenetworg/oauth2-azure": "Needed for Microsoft XOAUTH2 authentication"
},
"type": "library",
"autoload": {
  "psr-4": {
    "PHPMailer\\": "src/"
  }
},
"notification-url": "https://packagist.org/downloads/",
"license": [
  "LGPL-2.1-only"
],
"authors": [
  {
    "name": "Marcus Bonton",
    "email": "phpmailer@synchromedia.co.uk"
  },
  {
    "name": "Jim Jagielski",
    "email": "jimjag@gmail.com"
  },
  {
    "name": "Andy Prevost",
    "email": "codeworxtech@users.sourceforge.net"
  },
  {
    "name": "Brent R. Matzelle"
  }
],
"description": "PHPMailer is a full-featured email creation and transfer class for PHP",
"support": {
  "issues": "https://github.com/PHPMailer/PHPMailer/issues",
  "source": "https://github.com/PHPMailer/PHPMailer/tree/v6.9.3"
},
"funding": [
  {
    "url": "https://github.com/Synchro",
    "type": "github"
  }
],
"time": "2024-11-24T18:04:13+00:00"
},
"packages-dev": [],
"aliases": [],
"minimum-stability": "stable",
"stability-flags": {},
"prefer-stable": false,
"prefer-lowest": false,
"platform": {},
"platform-dev": {},
"plugin-api-version": "2.6.0"
}

```

composer.json

```
{
  "require": {
    "phpmailer/phpmailer": "^6.9"
  }
}
```

```

        }
    }

package.json

{
  "dependencies": {
    "cloudinary": "^2.6.0"
  }
}

package-lock.json

{
  "name": "CSE-316",
  "lockfileVersion": 3,
  "requires": true,
  "packages": {
    "": {
      "dependencies": {
        "cloudinary": "^2.6.0"
      }
    },
    "node_modules/cloudinary": {
      "version": "2.6.0",
      "resolved": "https://registry.npmjs.org/cloudinary/-/cloudinary-2.6.0.tgz",
      "integrity": "sha512-FlIy9RR5LPgkMloG4V7yUpC6ASyIFQMWfx4TgOi/xBeLxJTegebYQc3itiXLob0IDISaL0KyT2THEw6osrKqpQ==",
      "license": "MIT",
      "dependencies": {
        "lodash": "4.17.21",
        "q": "1.5.1"
      },
      "engines": {
        "node": ">9"
      }
    },
    "node_modules/lodash": {
      "version": "4.17.21",
      "resolved": "https://registry.npmjs.org/lodash/-/lodash-4.17.21.tgz",
      "integrity": "sha512-v2kDEe57ecTulaDiuNTPy3Ry4gLJ6Z1O3vE1krXZNsQ+LFTGHVxVjcXPs17LhbZVGedAJv8XZ1tvj5FvSg==",
      "license": "MIT"
    },
    "node_modules/q": {
      "version": "1.5.1",
      "resolved": "https://registry.npmjs.org/q/-/q-1.5.1.tgz",
      "integrity": "sha512-kV/CThKx06xyFEZUugw/+plOywXcDbFYgSct5cT3gqlbkBE1SJdwY6UQoZvodiWF/ckQLzyDE/Bu1M6gVu5IVw==",
      "deprecated": "You or someone you depend on is using Q, the JavaScript Promise library that gave JavaScript developers strong feelings about promises. They can almost certainly migrate to the native JavaScript promise now. Thank you literally everyone for joining me in this bet against the odds. Be excellent to each other.\n\n(For a CapTP with native promises, see @endo/eventual-send and @endo/captP)",
      "license": "MIT",
      "engines": {
        "node": ">0.6.0",
        "teleport": ">0.2.0"
      }
    }
  }
}

```