The **OTP.php** file demonstrates high cohesion by performing a single, clearly defined task: **verifying the one-time password (OTP)** submitted by the user. The script combines six input fields into a full code, reads user data from **users.json**, checks for a match, and then either redirects the user or shows an error message. All logic in this file is directly related to OTP verification — it does not handle password resetting, email sending, or registration. By keeping the responsibility limited and focused, the code becomes easier to understand, test, and maintain — which is a textbook example of high cohesion.

|  |
| --- |
| <?php  session\_start(); //for cookies  // Check if form was submitted  if ($\_SERVER["REQUEST\_METHOD"] === "POST") {      // Grab the 6 digits and combine them      $otp = $\_POST['otp1'] . $\_POST['otp2'] . $\_POST['otp3'] . $\_POST['otp4'] . $\_POST['otp5'] . $\_POST['otp6'];      // Load user data from JSON      $json\_file = '../Data/users.json';      $json\_data = file\_get\_contents($json\_file);      $json\_dec = json\_decode($json\_data, true);      $matchFound = false;      // Check if OTP matches any user      foreach ($json\_dec as $index => $user) {          if (isset($user['OTP']) && $user['OTP'] == $otp && $\_SESSION['email'] == $user['email']) {              $matchFound = true;              $json\_dec[$index]['OTP'] = null;              break;          }      }      //updating the users.json      $json\_en = json\_encode($json\_dec, JSON\_PRETTY\_PRINT);      file\_put\_contents($json\_file, $json\_en);      if ($matchFound){          header ('Location: ../phpfiles/Re-enablePassword.php');          //delete otp          exit();      } else {          header ('Location: ../htmlfiles/OTP.html?msg=invalid');          exit();      }  }  ?> |

The code used in **ForgotPassword.php** demonstrates **low cohesion** because it performs several unrelated tasks within the same block. It generates a 6-digit OTP, stores session data, formats and sends an email using PHPMailer, and updates the **users.json** file. These responsibilities — authentication logic, session handling, email composition, and file operations — should ideally be separated into distinct components or functions. Keeping them all together reduces maintainability and makes testing or modifying any single behaviour more error-prone. This is a clear example of **low cohesion**, where the script lacks a single, focused purpose.

|  |
| --- |
| $OTP = rand(100000, 999999); //creating a 6 digit code for authentication which will be sent to the user's email.      $OTP\_str = strval($OTP);     foreach ($dec\_data as $index => $user)      {          if($user['email'] === $email)          {              $dec\_data[$index]['OTP'] = $OTP\_str;              $\_SESSION['email'] = $user['email'];              break;          }        }      $mail->isHTML(true);  // Setting email format to HTML      $mail->Subject = 'Request to re-enable your password.';      $mail->Body = 'Please enter the 6 digit code provided to you, in the link sent, to re-enable your new password.<br>Code: '. $OTP;      // Send the email      $mail->send();      //updating the json file data      $json\_en = json\_encode($dec\_data, JSON\_PRETTY\_PRINT);      file\_put\_contents($json\_file, $json\_en);      //header("Location: ../htmlfiles/OTP.html");      exit();  } catch (Exception $e) {      echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";  } |