# Bazaar – php example code – part 12 – CAPTCHA login hardening

**Article focus on implementation of CAPTCHA with tools available in PHP. Separate script generate randomly rotated and by scratches and lines distorted text consist from 6 alphanumeric characters stored in a picture. Also provide hash of generated pass_phrase stored in session variable. Login page read text from verification field, hash them and compare against value stored in session. If password and captcha pass_phrases are as expected. User is validated as logged in user.**

## Expectation from CAPTCHA hardened login

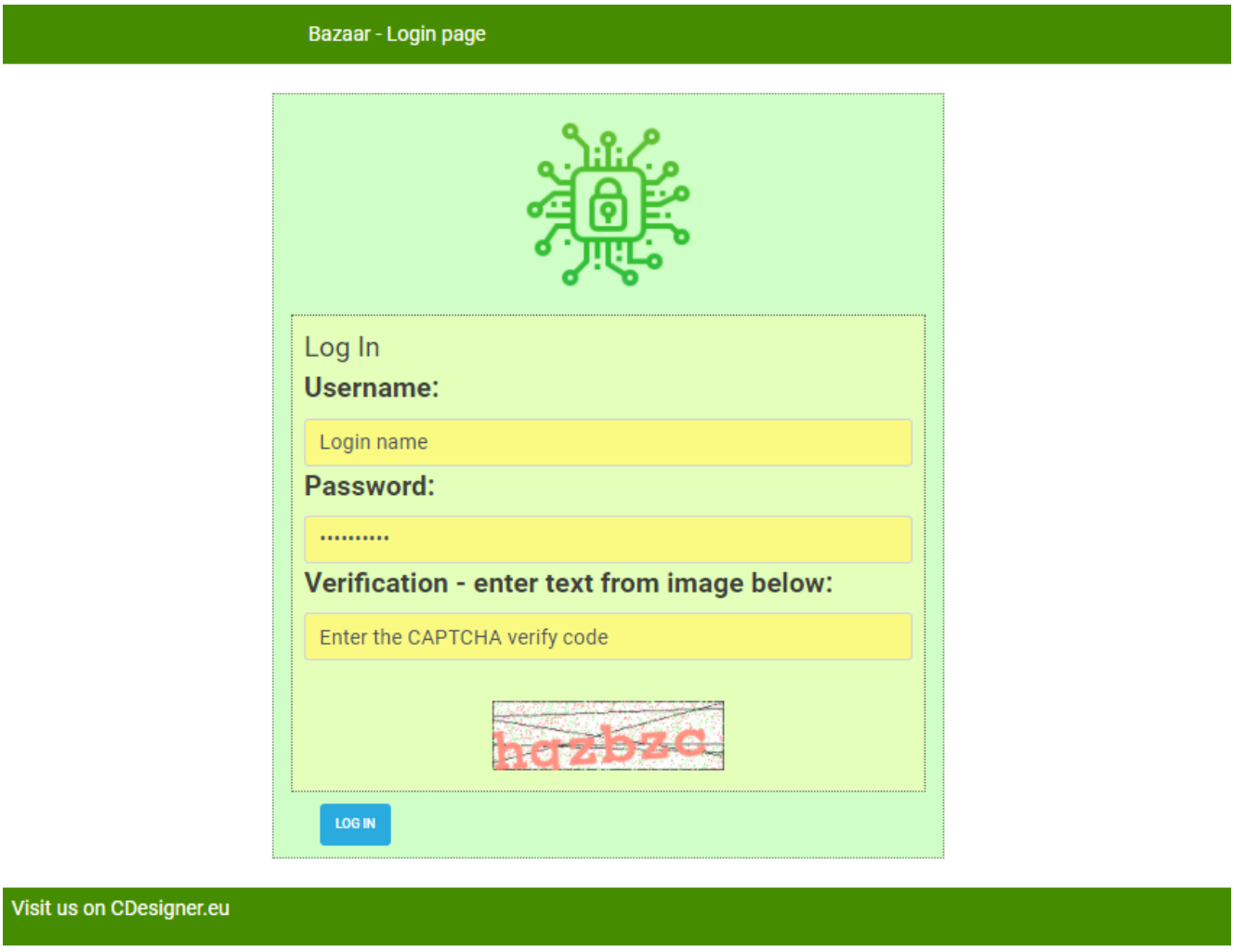
A **CAPTCHA** (/kæp.tʃə/, a contrived acronym for „Completely Automated Public Turing test to tell Computers and Humans Apart“) is a type of challenge–response test used in computing to determine whether or not the user is human. ( as is is mentioned in wiki, 25.12.2020).

Most common way how to implement captcha verification, is generating human readable picture containing alphanumeric characters that are displayed on pages with some type of prompt for a data. Our login page expect login name and password, rouge system (robot) can attempt for brute force access gain. Inserting third field for retyping code from provided image is a way how to eliminate automatized mechanism for gaining page access.

Our expectation from way how it will be implemented are:

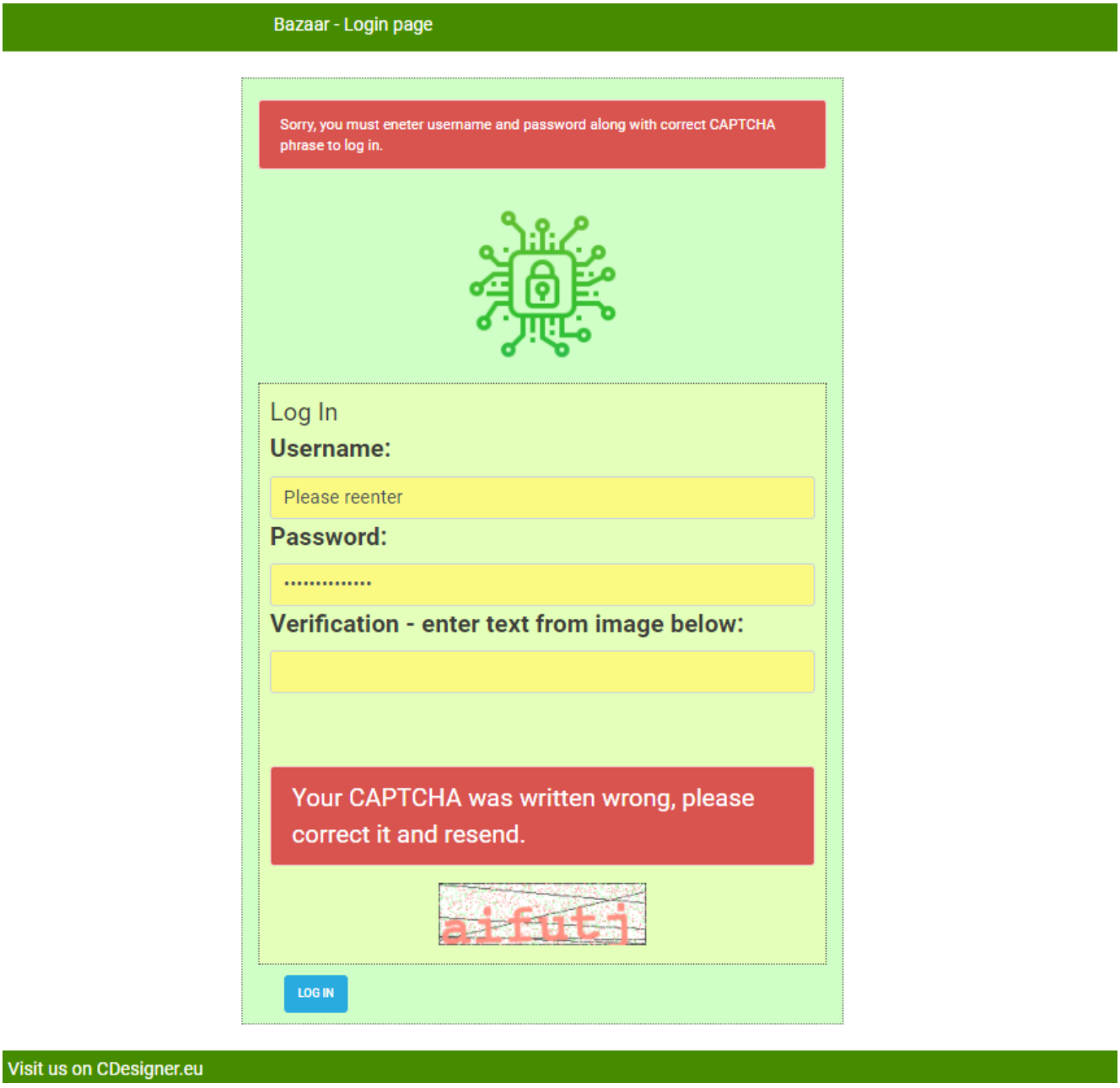
- simple CAPTCHA image generation only with resource available in PHP code
- generated CAPTCHas must be hardly readable by captcha OCR software (image rotation, scratches, ping or red color text, random lines)
- main captcha code stored separately from login or other pages implementing them
- way how to delete used captcha images
- quick implementation on login page without large code retyping

Captcha image is implemented in frontpage of login as it display next image.



Captcha verification on login page

If user type incorrect text, error message is displayed along with new CAPTCHA image as it is shown on next picture.




## captcha.php generating code

Our CAPTCHA images generating ode is stored in separate

script. This approach enable further improvements and transfer into other applications.

Leading part of code enable define dimensions of generated image, next variety of alphanumeric characters included in to a code and number of them in one image.

```
<!–
********************************************************************** –>
<!– PHP  code generating verification captcha image
                                 –>
<!–
********************************************************************** –>
<!– Vrsion: 1.0        Date: 8. – 9.11.2020 by CDesigner.eu
                                 –>
<!–
********************************************************************** –>
<?php
    //require_once(‚appvars.php‘); // including variables for database

    // if included whole not necessary session_start(); // start the session – must be added on all pages for session variable accessing
    // solution using SESSIONS with COOKIES for longer (30days) login persistency

    /*if(!isset($_SESSION[‚users_id‘])) { // if session is no more active
        if(isset($_COOKIE[‚users_id‘]) && isset($_COOKIE[‚username‘])) { // but cookie is set then renew session variables along them
            $_SESSION[‚users_id‘] = $_COOKIE[‚users_id‘];
            $_SESSION[‚username‘] = $_COOKIE[‚username‘];
            $_SESSION[‚user_role‘] = $_COOKIE[‚user_role‘]; // added for role
```

```
        }
    } */

    // important captcha constants
    define(,CAPTCHA_NUMCHARS‘, 6); // number of charakters in
 CAPTCHA
    define(,CAPTCHA_WIDTH‘, 200); // width of image
    define(,CAPTCHA_HEIGHT‘, 60); // height of image
    // Set Correct Path to Font File
    $fontPath=’C:\xampp_7_4_2020\htdocs\bazaar\images\courier
_new_bold.ttf‘;
    // generating passphrase by random numbers
    $pass_phrase = „“;
    for($i = 0; $i < CAPTCHA_NUMCHARS; $i++ ) {
       $pass_phrase .= chr(rand(97, 122));
    }
    // store the encryption pass-phrase in a session variable
    $_SESSION[,pass_phrase‘] = sha1($pass_phrase);
    //create the image
    $img = imagecreatetruecolor(CAPTCHA_WIDTH, CAPTCHA_HEIGHT
);
    //set a white background with black text and gray graphic
s
    $bg_color = imagecolorallocate($img, 255, 255, 255); //wh
ite
    $text_color = imagecolorallocate($img, 255, 146, 130); //
pale red
    $graphic_color = imagecolorallocate($img, 64, 64, 64); //
darkgray
    $graphic_color_noise_red = imagecolorallocate($img, 255,
128, 128); //red noise pattern
    $graphic_color_noise_green = imagecolorallocate($img, 128
, 255, 128); //green noise pattern
    // fill the background
    imagefilledrectangle($img, 0, 0, CAPTCHA_WIDTH, CAPTCHA_H
EIGHT, $bg_color);
    // image edges rectangle drawing
    imagerectangle ( $img , 0 , 0, CAPTCHA_WIDTH -1  , CAPTCH
A_HEIGHT -1 , $graphic_color  );
    //draw some random lines
    for($i = 0; $i < 5; $i++) {
```

```
         imageline($img,0, rand() % CAPTCHA_HEIGHT, CAPTCHA_WI
DTH, rand() % CAPTCHA_HEIGHT, $graphic_color);
     }

     //sprinkle in some random green dots
     for($i = 0; $i < 1000; $i++) {
        imagesetpixel($img, rand() % CAPTCHA_WIDTH,  rand() %
CAPTCHA_HEIGHT, $graphic_color_noise_green);
    }
    // draw the pass-phrase string
    imagettftext($img, 36, rand(0,10), rand(0, 12) , CAPTCHA_H
EIGHT – rand(-5, 5), $text_color, $fontPath, $pass_phrase);
    //sprinkle over in some random dots
    for($i = 0; $i < 1000; $i++) {
        imagesetpixel($img, rand() % CAPTCHA_WIDTH,  rand() %
CAPTCHA_HEIGHT, $graphic_color_noise_red);
    }

    // VERY IMPORTANT: Prevent any Browser Cache!! – older app
roach send by header
    // header(„Cache-Control: no-store,
    //no-cache, must-revalidate“);
    // output the image as PNG using a header;
    /* ob_clean();
    header(„Content-type: image/jpg“);
    imagejpg($img);*/
    // creating filename and sending them through session and
variable
    $imageCaptchafilename = IMAGE_PATH . „captcha“.rand(1,1000
).“.png“;
    // debug echo $imageCaptchafilename;
    $_SESSION[,imageCaptchafilename‘] = $imageCaptchafilename;
    //writting image to png
    imagepng($img, $imageCaptchafilename, 5);
    //clean up
    imagedestroy($img);
 ?>
```

Example of generated image for closer look follows


## Improved login page with CAPTCHA

Link to generated CAPTCHA image and pass_phrase is available in session variables for login page scripts.

Existing form code is extended for verification field and is followed by CAPTCHA image. After unsuccessfully retyped code, error message is displayed formatted with bootstrap danger style.

Al parts implementing CAPTCH in login page are marked by orange for better understanding and distinguishing them from other text.

```
<!–
********************************************************************** –>
<!– PHP „self“ code handling login into the bazaar app
                                 –>
<!–
********************************************************************** –>
<!– Vrsion: 1.0         Date: 11.10-24.10.2020 by CDesigner.eu
                                 –>
<!–
********************************************************************** –>
<?php
 require_once(‚appvars.php‘); // including variables for datab
ase
```

```
 require_once(‚captcha.php‘); // including generator of captch
a image
 session_start(); // start the session

 // two variables for message and styling of the mesage with b
ootstrap
 $msg = “;
 $msgClass = “;
 $usr_username = “;
 $usr_passwd = “;
 $verified_human_by_CAPTCHA = -1; //
//get info that user is loged in, if not try it looking at coo
kies
//if(!isset($_COOKIE[‚s‘])) { old solution with cookies
  if(!isset($_SESSION[‚users_id‘])) { //new with session varia
bles
    if(isset($_POST[‚submit‘])) {
        /* Attempt MySQL server connection.  */
             $dbc = mysqli_connect(DB_HOST, DB_USER, DB_PW, DB
_NAME);

               // accessing user entered login data
             $usr_username = htmlspecialchars($_POST[‚u_name‘]
);
             $usr_passwd = htmlspecialchars($_POST[‚u_pass‘]);
             //implement CAPTCHA pass-phrase verification

            $user_pass_phrase = sha1(htmlspecialchars($_POST[‚
verify‘]));
            $pass_phrase_now = htmlspecialchars($_POST[‚pass_p
hrase_now‘]);
            $imageCaptchafilename_now = htmlspecialchars($_POS
T[‚imageCaptchafilename_now‘]); // name of current captcha pho
to file for deletion after usage

           if($pass_phrase_now == $user_pass_phrase) {
             $verified_human_by_CAPTCHA = 1;
             @unlink($imageCaptchafilename_now); // delete ca
ptcha file
               //debug echo „captcha ok“;
```

```
            } else {
              $verified_human_by_CAPTCHA = 0;
              @unlink($imageCaptchafilename_now); // also dele
te captcha file because new one was created
              $msgClass = ‚alert-danger‘;
              $msgCAPTCHA = „Your CAPTCHA was written wrong, p
lease correct it and resend.“;
            };
             if(!empty($usr_username) && !empty($usr_passwd) &
& $verified_human_by_CAPTCHA) {
              // try lookup user database
              $usr_passwd_SHA = sha1($usr_passwd);
              $sql = „SELECT users_id, username, user_role FRO
M bazaar_user WHERE username = „.“,$usr_username'“. “ AND pass
_word = „.“,$usr_passwd_SHA'“ ;
              // debug output echo  $usr_username;
              // echo  $usr_passwd;
              //echo $usr_passwd_SHA;
              $data = mysqli_query($dbc, $sql);

              if(mysqli_num_rows($data) == 1) {
                  // login is ok, set user  ID and username co
okies and redirect to the homepage
                 $row = mysqli_fetch_array($data);
                 //setcookie(‚users_id‘, $row[‚users_id‘]); o
ld solution with cookies
                 //setcookie(‚username‘, $row[‚username‘]);
                 $_SESSION[‚users_id‘] = $row[‚users_id‘]; //
 sloution with sessions
                 $_SESSION[‚username‘] = $row[‚username‘];
                 $_SESSION[‚user_role‘] = $row[‚user_role‘];
// added user_role session variable
                 // new cookies for login persistency that ex
pires after 30 days without logout combination SESSION with CO
OKIES is awailable
                 setcookie(‚users_id‘, $row[‚users_id‘], time
()+(60+60*24*30));
                 setcookie(‚username‘, $row[‚username‘], time
()+(60+60*24*30));
                 setcookie(‚user_role‘, $row[‚user_role‘], ti
me()+(60+60*24*30)); // cookie for user_role of loged in user
```

```
added
                    $home_url = ,http://‘. $_SERVER[,HTTP_HOST‘]
 . dirname($_SERVER[,PHP_SELF‘]) . ,/index.php‘;
                    header(,Location:‘. $home_url);
                    // Free result set
                    mysqli_free_result($data);
                    // Close connection
                    mysqli_close($dbc);
                } else  {
                    // urename/ password are incorrect – error m
eesage is displayed
                    $msg = „Incorrect username or password. Logi
n denied!  „;
                    $msgClass = ,alert-danger‘;

            }

            } else {
                // username/ password were not entered – displ
ay error message
                $msg = „Sorry, you must eneter username and pa
ssword along with correct CAPTCHA phrase to log in. „;
                $msgClass = ,alert-danger‘;

            }
    }
}
?>
<!– ****************************************** –>
<!– HTML code containing Form for submitting –>
<!– ****************************************** –>
<!DOCTYPE html>
<html>
<head>
  <title> Bazaar login page  </title>
  <link rel=“stylesheet“ href=“./css/bootstrap.min.css“> <!– b
ootstrap mini.css file –>
  <link rel=“stylesheet“ href=“./css/style.css“> <!– my local.
css file –>
    <script src=“https://code.jquery.com/jquery-3.1.1.slim.min
.js“                                integrity=“sha384-
```

```
A7FZj7v+d/sdmMqp/nOQwliLvUsJfDHW+k9Omg/a/EheAdgtzNs3hpfag6Ed95
0n“ crossorigin=“anonymous“></script>
        <script src=“https://cdnjs.cloudflare.com/ajax/libs/te
ther/1.4.0/js/tether.min.js“      integrity=“sha384-
DztdAPBWPRXSA/3eYEEUWrWCy7G5KFbe8fFjk5JAIxUYHKkDx6Qin1DkWx51bB
rb“ crossorigin=“anonymous“></script>

</head>
<body>
  <nav class=“navbar „>
      <div class=“container“ id=“header_container_580″>
        <div class=“navbar-header“>
          <?php
             require_once(,headerlogo.php‘);
          ?>
                                      <a    class=“navbar-
brand“  href=“index.php“>Bazaar – Login page</a>
        </div>
      </div>
    </nav>
    <div class=“container“ id=“formcontainer“>
    <?php if($msg != “): ?>
        <br>
        <div class=“alert <?php echo $msgClass; ?>“><?php echo
 $msg; ?></div>
     <?php endif; ?>

     <?php
           //if(empty($_COOKIE[,users_id‘])) { solution with
cookies
             if(empty($_SESSION[,users_id‘])) { // solution w
ith sessions
               // only show for if session with name users_id
 does not exist
               //echo , <br> ,;
                     //echo   ,<p class=“alert alert-
danger“>‘ . $msg . ,</p>‘;
      ?>

       <br>
       <img id=“calcimage“ src=“./images/login.png“ alt=“baza
```

```
ar image“ width=“150″ height=“150″>
        <br>
        <form  method=“post“ action=“<?php echo $_SERVER[,PHP_
SELF‘]; ?>“>
           <div id=“login“>
                <legend> Log In <legend>
                <label>Username:</label>
                    <input type=“text“ onfocus=“this.value='<?
php echo isset($_POST[,u_name‘]) ? “ : “; ?>'“ name=“u_name“ c
lass=“form-
control“ value=“<?php echo isset($_POST[,u_name‘]) ? ,Please r
eenter‘ : ,Login name‘; ?>“>
                    <label>Password:</label>
                    <input type=“password“ onfocus=“this.value
='<?php echo isset($_POST[,u_pass‘]) ? “ : “; ?>'“ name=“u_pas
s“                                                  class=“form-
control“ value=“<?php echo isset($_POST[,u_pass‘]) ? ,Please r
eenter‘ : ,Login name‘; ?>“>
                    <label for=“verify“>Verification – enter t
ext from image below:</label>
                    <input type=“text“ onfocus=“this.value='<?
php echo isset($_POST[,verify‘]) ? “ : “; ?>'“ name=“verify“ c
lass=“form-
control“ value=“<?php echo isset($_POST[,verify‘]) ? “ : ,Ente
r the CAPTCHA verify code‘; ?>„>
                    <br>
                    <?php if(($verified_human_by_CAPTCHA == 0)
 ): //error messaging if wrong CAPTCHA?>
                    <br>
                    <div class=“alert <?php echo $msgClass; ?>
“><?php echo $msgCAPTCHA; ?></div>
                    <?php endif; ?>
                    <center> <img src=“<?php echo $imageCaptch
afilename ; ?>“ alt=“Verification pass-phrase“ > </center>
                    <!– ass a hidden is sent sha actualy gener
ated                    captcha                         pass-
phrase only this way it is producet in same run –>
                    <input type=“hidden“ name=“pass_phrase_now
“ value=“<?php echo sha1($pass_phrase); ?>“ />
                    <!– as a hidden is sentname of captcha fil
e for deletion after use –>
```

```
                    <input type=“hidden“ name=“imageCaptchafil
ename_now“ value=“<?php echo $imageCaptchafilename; ?>“ />
             </div>
            <input id=“loginsubmitt“ type=“submit“ name=“submit
“ class=“btn btn-info“ value=“Log In“>
           <br>
        </form>
        <?php }  else {
                // successfull login
                 // cookie solution echo ,<p class=“alert ale
rt-
success“> You are loged in as , . $_COOKIE[,username‘]. ,</p>‘
;
                 echo ,<br>‘;
                          echo  ,<p  class=“alert  alert-
success“> You are loged in as <em>‘ . $_SESSION[,username‘]. ,
</em></p>‘; // session solution
                          echo  ,<p  class=“alert  alert-
success“> If you will logout or login with anither credentials
, please first <a href=“logout.php“>logout!. </a></p>‘;
              }
        ?>
      </div>



      <?php  // footer include code
          require_once(,footer.php‘); // including footer
          generate_footer(580); // function from footer.php fo
r seting width, you can use 580 and 1060px width
        ?>

</body>
</html>
```

As it was mentioned before we must have way how to signal generated codes for verification, they are stored in session variable as pass_phrase, also image name. Second problem is, how to deleted unused images. Our implementation is simple, but if app run with many connected clients then in a short time can by generated many pictures. There is small posibility for generate one with same numbering part. Wider number is for

consideration, also new problems will arisen in heavy loads that we must take in mind.

## Conclusion

CAPTCHA verified login provide new level of security for gaining access to our application. Please keep in mind, that verification user and distinguishing them for automatized scripts is now a must have thing. It is a bare minimal for supporting basic level of security for today apps.

Full code of our bazaar education project can be obtained from github here.