```
/*--- lib_auth.php ---*/
/*
<?php

  ////////////////////////////////////////////////////////////////////////////////
  // Authentication Version 4.0 - Template - J. Kenney 2016-2018

  // Load Configuration
  require_once(’../etc/config.inc.php’);

  // Load MySQL library (must provide $db object)
  if (defined(’LIBRARY_PATH’)) {
    require_once(LIBRARY_PATH.’lib_mysql.php’);
  } else {
    require_once(’lib_mysql.php’);
  }

  ////////////////////////////////////////////////////////////////////////////////
  // Start the session and retrieve the session_id
  session_start();
  define(’AUTH_SESSION_ID’, session_id());

  ////////////////////////////////////////////////////////////////////////////////
  // Validate current status (are we still correctly logged in)
  function authentication_verify($db) {
    $query = "SELECT user, session, fullname, department, first, last
                FROM auth_user
                     LEFT JOIN auth_session USING(user)
               WHERE (NOW() < (DATE_ADD(lastvisit, INTERVAL ".AUTH_MAX_TIME_IDLE_SE
SSION.")))
                 AND (NOW() < (DATE_ADD(lastlogin, INTERVAL ".AUTH_MAX_TIME_SINCE_L
AST_LOGIN.")))
                 AND auth_session.id=?";
    $stmt = build_query($db, $query, array(AUTH_SESSION_ID));
    $stmt->store_result();
    $USER_CREDENTIALS = array();
    if ($stmt->num_rows > 0) {
      $USER_CREDENTIALS = array_2d_to_1d(stmt_to_assoc_array($stmt));
    }
    $stmt->close();
    return $USER_CREDENTIALS;
  }

  function authentication_redirect($db) {
    // Which Authentication Script should we run?
    if (defined(’DEVELOPER_HOSTNAME’)) {
      $dev_flag = False;
      foreach(DEVELOPER_HOSTNAME as $i => $host) {
        if (gethostname() == $host) {
          $dev_flag = True;
        }
      }
      if ($dev_flag) {
        require_once(AUTH_DEV_LIBRARY);
      } else {
        require_once(AUTH_LIBRARY);
      }
    } else {
      require_once(AUTH_LIBRARY);
    }
    if (isset($USER_CREDENTIALS)) {
      return $USER_CREDENTIALS;
    }
    return array();
  }

  function authentication_logoff($db) {
    // Build delete user query and execute
```

```
    $query = "DELETE FROM auth_session WHERE id=?";
    $stmt = build_query($db, $query, array(AUTH_SESSION_ID));
    $stmt->close();

    // Destroy the current session information (force logoff)
    session_destroy();
    session_start();

    // Blank out the query_string assuming a login redirect...
    $_SERVER[’QUERY_STRING’] = ’’;
  }

  function authentication_update($db, $USER_CREDENTIALS) {
    $query = "INSERT INTO auth_session (user, id, lastvisit) VALUES (?, ?, NOW())
              ON DUPLICATE KEY UPDATE user=?, lastvisit=NOW()";
    $stmt = build_query($db, $query, array($USER_CREDENTIALS[’user’],
                                           AUTH_SESSION_ID,
                                           $USER_CREDENTIALS[’user’]));
    $stmt->close();
  }

  function authentication_login($db, $USER_CREDENTIALS) {
    // Add user information to database session table
    $query = "INSERT INTO auth_user (user, fullname, first, last, department, lastl
ogin) VALUES (?,?,?,?,?,NOW())
              ON DUPLICATE KEY UPDATE fullname=?, first=?, last=?, department=?, la
stlogin=NOW()";
    $stmt = build_query($db, $query, array($USER_CREDENTIALS[’user’],
                                           $USER_CREDENTIALS[’fullname’],
                                           $USER_CREDENTIALS[’first’],
                                           $USER_CREDENTIALS[’last’],
                                           $USER_CREDENTIALS[’department’],
                                           $USER_CREDENTIALS[’fullname’],
                                           $USER_CREDENTIALS[’first’],
                                           $USER_CREDENTIALS[’last’],
                                           $USER_CREDENTIALS[’department’]));
    $stmt->close();

    // Store the results in the session
    $_SESSION[’user’] = $USER_CREDENTIALS;

    // Prevent accidental reloads of this portion
    if (isset($_REQUEST[’token’])) {
      $_SESSION[’token’] = $_REQUEST[’token’];
    }

  }

  ////////////////////////////////////////////////////////////////////////////////
  // If requested, logoff the user
  if (isset($_REQUEST[’logoff’])) {
    authentication_logoff($db);
  }

  ////////////////////////////////////////////////////////////////////////////////
  // Has authentication information been received?  If so validate or redirect
  if (   isset($_REQUEST[’token’])
      && isset($_REQUEST[’user’])
      && isset($_REQUEST[’date’])
      && (  !isset($_SESSION[’token’])
                || $_SESSION[’token’] != $_REQUEST[’token’])
     ) {

    $USER_CREDENTIALS = authentication_redirect($db);
    authentication_login($db, $USER_CREDENTIALS);
    authentication_update($db, $USER_CREDENTIALS);
  } else {
    $USER_CREDENTIALS = authentication_verify($db);
    if (empty($USER_CREDENTIALS)) {
      unset($USER_CREDENTIALS);
    } else {
      authentication_update($db, $USER_CREDENTIALS);
    }
  }

  ///////////////////////////////////////////////////////////////
  // So, are we currently logged in?
  if (isset($USER_CREDENTIALS) && !empty($USER_CREDENTIALS)) {
  } elseif (isset($_REQUEST[’login’]) || !isset($MODULE_DEF[’guest’]) || !$MODULE_D
EF[’guest’]){
    $USER_CREDENTIALS = authentication_redirect($db);
    authentication_login($db, $USER_CREDENTIALS);
    authentication_update($db, $USER_CREDENTIALS);
  } elseif (isset($MODULE_DEF[’guest’]) || $MODULE_DEF[’guest’]) {
    $USER_CREDENTIALS = array(’user’          => ’guest’,
                              ’fullname’      => ’Guest’,
                              ’first’         => ’Guest’,
                              ’last’          => ’Guest’,
                              ’department’    => ’Guest’,
                              ’time’          => 0,
                              ’privileges’    => array());
  }

  ///////////////////////////////////////////////////////////////
  // Safeguard, did we make it here without authentication?
  if (!isset($USER_CREDENTIALS)) {
    echo "<hr><h1><font color=red>Fatal Error (Authentication Safeguard), Contact A
dministrator</font></h1>";
    die;
  }

  ///////////////////////////////////////////////////////////////
  // Do we want to become a different user?
  $query = "SELECT access, value FROM auth_access WHERE user=? AND access=’admin’ A
ND value=’become’";
  $stmt = build_query($db, $query, array($USER_CREDENTIALS[’user’]));
  $stmt->bind_result($returned_access, $returned_value);
  $stmt->store_result();
  if ($stmt->num_rows > 0) {
    $stmt->close();
    if (defined(’AUTH_BECOME_LIBRARY’) && isset($_REQUEST[’become’])) {
      require_once(AUTH_BECOME_LIBRARY);
      authentication_login($db, $USER_CREDENTIALS);
      authentication_update($db, $USER_CREDENTIALS);
    }
  } else {
    $stmt->close();
  }

  ///////////////////////////////////////////////////////////////
  // Retrieve Access Information
  $query = "SELECT access, value FROM auth_access WHERE user=?";
  $stmt = build_query($db, $query, array($USER_CREDENTIALS[’user’]));
  $stmt->bind_result($returned_access, $returned_value);
  $USER_CREDENTIALS[’privileges’] = array();
  while($stmt->fetch()) {
    $USER_CREDENTIALS[’privileges’][$returned_access][] = $returned_value;
  }
  $stmt->close();

  ///////////////////////////////////////////////////////////////
  // Determine Admin Status
  if (isset($USER_CREDENTIALS[’privileges’][’admin’])) {
    define(’ADMIN’, True);
  } else {
    define(’ADMIN’, False);
  }
```

```
  ///////////////////////////////////////////////////////////////
  // Are they an instructor/staff member?
  // Search for m123456 <- and do not promote to instructor
  if ($USER_CREDENTIALS[’user’] == ’guest’) {
    define(’GUEST’, True);
    define(’STUDENT’, False);
    define(’INSTRUCTOR’, False);
  } elseif (preg_match("/^M[0-9]{6}/", $USER_CREDENTIALS[’user’]) == 0
         && preg_match("/^m[0-9]{6}/", $USER_CREDENTIALS[’user’]) == 0) {
    define(’GUEST’, False);
    define(’STUDENT’, False);
    define(’INSTRUCTOR’, True);
  } else {
    define(’GUEST’, False);
    define(’STUDENT’, True);
    define(’INSTRUCTOR’, False);
  }

  ///////////////////////////////////////////////////////////////
  // Are they an Officer? USMC? USN? USAF? USA? Midn? Civilian?
  // Set $user_type appropriately;
  foreach (array(’ USMC ’=>’MARINE OFFICER’, ’ USN ’ =>’NAVY OFFICER’,
                 ’ USA ’ =>’ARMY OFFICER’,   ’ USAF ’=>’AIR FORCE OFFICER’,
                 ’ CIV ’ =>’CIVILIAN’,
                 ’ Midn ’=>’MIDN’,           ’ MIDN ’=>’MIDN’)
           as $search => $report) {
      if (strpos($USER_CREDENTIALS[’fullname’], $search) !== false && !defined(’TYP
E’)) {
        define(’TYPE’, $report);
      }
  }
  if (!defined(’TYPE’)) {
    define(’TYPE’, ’UNKNOWN’);
  }

  ///////////////////////////////////////////////////////////////
  // Provide back the USER constant to any calling pages.
  define(’USER’, $USER_CREDENTIALS);
  unset($USER_CREDENTIALS);

  ///////////////////////////////////////////////////////////////
  // SECURITY PROTOCOL based on $MODULE_DEF
  $visible = True;
  if (isset($MODULE_DEF) && !empty($MODULE_DEF)) {
    if (STUDENT && !$MODULE_DEF[’student’]) { $visible = False; }
    if (INSTRUCTOR && !$MODULE_DEF[’instructor’]) { $visible = False; }
    if (GUEST && !$MODULE_DEF[’guest’]) { $visible = False; }
    foreach ($MODULE_DEF[’access’] as $key => $value) {
      if (!isset(USER[’privileges’][$key]) || !in_array($value, USER[’privileges’][
$key])) {
        $visible = False;
      }
    }
  } else {
    $visible = False;
  }

  ///////////////////////////////////////////////////////////////
  // Fail Secure if blocked by security
  if (!$visible) {
    echo "<h1>You are not authorized to view this page</h1>";
    echo "<p>Please contact the system administrator.</p>";
    die;
  }


?>
\n*/
```