

How do we manage logins?

ICS4U – Mr. Emmell

Logins revolve around two functions

- `password_hash($passwordToHash, PASSWORD_DEFAULT);`
 - Returns a string
 - Returned value is the hashed password to store in DB
- `Password_verify($passwordToCheck, $existingHash);`
 - Returns a Boolean
 - Returned value will be true if password matches the hashed value

4 Usecases for logins

- Check for a valid password
- Sign in the current user
- Check if the current user is signed in
- Log the user out

Check for a valid password

- Make a normal HTML form
 - Be sure to use type="password" for the password field
- Receive the login information and check it against what is contained in the database:

```
if (password_verify($password, $hash)) {  
    // Success!  
}  
else {  
    // Invalid credentials  
}
```

Sign in the current user

- FIRST!
 - We need to understand the “\$_SESSION” variable....

The \$_SESSION variable

- \$_SESSION is a magic variable that stays persistent for the entire duration that a given browser is on a given site. Yes, this means across different pages.
- If something is placed into this array, you will be able to access it on other pages.
- To enable use of sessions, must run `session_start();`

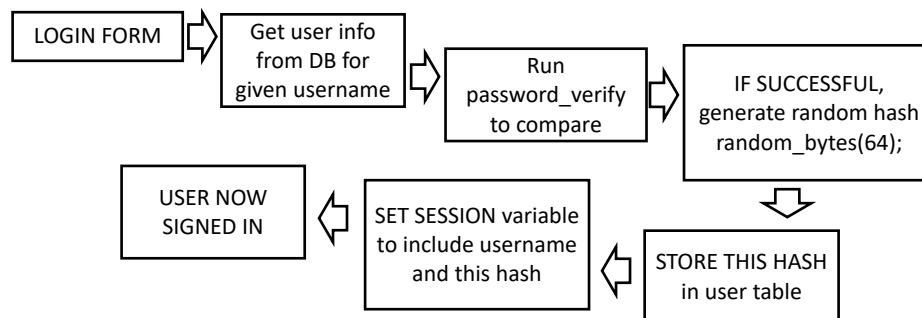
\$ SESSION Example

```
<?php
    session_start();
    if (//... successful login ...//) {
        $_SESSION['userID'] = $userID;
    }
}
```

- This data will now be accessible on other pages
 - (see \$_SESSION examples on Mr. Emmell's example website)

Sign in the current user

- Now...
- Basic flow is this:



Check if the current user is signed in

- Now that we know there is a hash in the user's session variable, and a matching hash in the database – simply compare the two

```
if (isset($_SESSION['username'], $_SESSION['hash'])) {  
    //get user record from MySQL  
    //Compare hash to existing $_SESSION['hash']  
    if ($row['hash'] == $_SESSION['hash']) {  
        //user is good and authenticated!  
    }  
    else {  
        //unset username and hash  
        //bounce to login page  
    }  
}
```

Log the user out

- A few simple steps

```
unset($_SESSION['user']);  
unset($_SESSION['hash']);  
header("location: index.php");
```

USE FUNCTIONS!

```

getUserInfo ($username)
//returns an array for the given user if exists

auth_checkLoginCredentials ($username, $password)
//returns true/false for the given username/password
//Intended to be used during the login process
//uses "getUserInfo" and "password_verify" (NO SQL)

auth_sessionHashCheck ($username, $userhash)
//checks the stored hash on the user's computer with the saved
// hash in the database. Returns T/F
// uses "getUserInfo" (NO SQL)

```

USE FUNCTIONS!

```

auth_genHash () {
    return random_bytes (128) ;
}
//generates the random session hash to be saved per student (NO SQL)

auth_saveHash ($username, $newHash)
//saves the given new hash into the database for the given user

auth_pageCheck ()
//returns true/false if user is already in. Intended to be used as the
quick check on all internal pages.
// Uses "auth_sessionHashCheck" function (NO SQL)

```

And now on every page...

```
<?php //at the top of the page
require("functions.php");

session_start();
if (!auth_pageCheck()) {
    header("location: logout.php");
}
?>
```