PIC 40A: Homework 7 (due 3/6 at 10pm)

Like on previous homeworks, it is important that you meet the following requirements.

- You must upload your files to **Gradescope** before the deadline.
- You must upload your files to the **PIC server** in the appropriate directory before the deadline.
- Both submissions must be **identical** (down to the character).
  Never make changes to the PIC server submission after the deadline.
  (We can see when a file was last modified.)
- You must tell us (me and the grader) your **PIC username**.
- You must **validate** your HTML using `https://validator.w3.org/`.
  Ideally, with PHP, you should check that, after removing the PHP parts, the remaining HTML validates.

In this assignment you will submit eight files...

1. `README.txt`. This will contain your PIC username.
2. `h_password.txt`.
   These is a completely new file that you will create in this homework.
3. `login.php`, `index.php`, `merch.php`, and `blog.php`.
   These files will build upon those that you submitted on a previous homework.
4. `username.js`. This is a file you made on a previous homework. You will remove all the code but the `get_username` function. `index.php` will no longer source this javascript file.
5. `login.js`. These is a file that you made on a previous homework. One line will change in here.

As mentioned above, you should submit all files to Gradescope before the deadline.
You should also submit the files to the PIC server. Save them in the directory

`/net/laguna/???...???/your_username/public_html/HW7`

(in the folder `HW7` within `public_html`). **You'll also need to create a folder called "sessions" with 755 permissions.** We should all be able to test your live webpage at

`www.pic.ucla.edu/~your_username/HW7`

Now, I am just left to tell you what I want these files to achieve. See the next page!

New Features

You are going to make it so your website has a secure login page. If the user tries to go anywhere on your website, they will be redirected to the login page until they've entered the correct password and a valid username. You will also add a secure hashed password.

login.php, login.js, and h_password.txt

So, in terms of the HTML...

- `login.php` will look identical to `login.html` but there will be a few changes to the HTML that are related to the PHP.

In terms of the PHP...

- You should use a web form (with method set to `POST`) for the password submission.
- The correct password should be: `Immutable`
  - A hashed version of the password should be stored in `h_password.txt`.
  - If the correct password is submitted, a PHP session with the name `myWebpage` should record the user as being `loggedin`, and the user could be redirected to `index.php`.
  - If an incorrect password is submitted, a PHP session with the name `myWebpage` should record the user as **not** `loggedin`, the user should stay on the same page and see a message saying `Invalid password!`.
- Copy and paste your previous homework submission's `login.js`. A few characters need editing to be consistent with using `php` file extensions instead of `html` file extensions.
- A proper username has to be submitted as well to be redirected. If the username validates and the password is correct, the user will be redirected to `index.php`.
- If the username doesn't validate, then the expected alert is thrown and the page is not redirected.

username.js

- This will only be sourced by `login.php` so you can remove everything but the `get_username` function definition.

index.php

- Copy your previous homework submission `index.html` and rename it as `index.php`. Then
  - use PHP sessions so that if a user is not logged in, they are redirected back to `login.php`;
  - use PHP cookies so that if a user is logged in but has not specified a username, they are redirected back to `login.php`;
- Use PHP so the user will see a greeting "Hello, {}!" at the top of the page. Where {} is the username they typed in the username text box of `login.php`.
- You will no longer source any JavaScript for this page.

merch.php

- Copy your previous homework submission `merch.html` and rename it as `merch.php`. Then
  - use PHP sessions so that if a user is not logged in, they are redirected back to `login.php`;
  - use PHP cookies so that if a user is logged in but has not specified a username, they are redirected back to `login.php`;
- `merch.js` can be included in your submission but its functionality won't be checked.

blog.php

- Copy your previous homework submission `blog.php`. Then
  - use PHP sessions so that if a user is not logged in, they are redirected back to `login.php`;
  - use PHP cookies so that if a user is logged in but has not specified a username, they are redirected back to `login.php`;
- `post.php` can be included in your submission but its functionality won't be checked.

Grading

- (1 point) The webpage `login.php` looks like `login.html` (and `logim.html`).
- (1 point) PHP sessions are named correctly and password submissions are appropriately hashed.
- (3 points) Before logging in or entering a username, `index.php`, `merch.php`, and `blog.php` will redirect to `login.php`.
- (3 points) Suppose no username is set but a valid password has been entered, `index.php`, `merch.php`, and `blog.php` will redirect to `login.php`.
- (3 points) Suppose a valid username is entered but a valid password has not been entered, `index.php`, `merch.php`, and `blog.php` will redirect to `login.php`.
- (1 point) An incorrect password submission on `login.php` results in staying on the page and seeing the message `Invalid password!`. Additionally...
  - (1 point) If there's an invalid username submission, you see the alert with the appropriate message.
  - (1 point) If there's a valid username submission, then the username is stored and fills the username textbox.
- (1 point) A correct password and invalid username submission results in an alert with the appropriate message. There should no message saying `Invalid password!`.
- (1 point) A correct password and valid username submission on `login.php` results in being redirected to `index.php`.
- (2 points) After being redirected to `index.php` you should see a greeting with the user's username.
- (1 point) `username.js` only contains the function `get_username` and is not sourced by `index.php`.
- (1 point) After logging in, entering an incorrect password at `login.php` results in being logged out, so that one of the other three pages redirects to `login.php` again.