Distributed Client/Server Programming
Lab 4

Objectives:

- Use an HTML form to gather username / password information from website visitors
- Use PHP and MySQL to check the username / password values
- Use PHP session values to store
  - if someone is logged in
  - what type of user (User or Admin) is logged in
- Secure the web pages so that Users cannot view Admin pages and vice versa

Instructions:

This week's lab will focus on using PHP sessions to secure a website. Our incredibly complex Great Web Application will consist of 2 content pages (1 for Users and 1 for the Admin) along with a login page and a logout page. Visitors to the website should not be able to view the content pages unless they are currently logged in with valid credentials.

In the lab instructions, the word "user" starting with a lowercase 'u' represents a website visitor (either a User or an Admin).

First, use the PHP script `setup_lab4.php` to create a table of users in your database for this lab. The table contains information (first name, last name, type of user, username, and password) for 2 Users and 1 Admin. It also creates a table of orders that each user has ordered from our theoretical e-commerce site.

Then edit the remaining files

```
login_page.php
user_page.php
admin_page.php
logout_page.php
```

If a user does not enter a valid username / password combination, they should receive an error message and the form should remain populated with their input. It is not necessary to ensure that the fields are populated before form submission.

Once a user presents a valid username / password pair they are forwarded to the correct content page (`user_page.php` for Users and `admin_page.php` for the Admin). To facilitate tracking if someone is logged in and what type of user they are, I suggest storing at least the "type" and "username" of the user who is currently logged in to the website in a PHP session.

If a user is logged in and they attempt to go to the login_page.php again, they should be forwarded to their content page (`user_page.php` for Users and `admin_page.php` for the Admin).

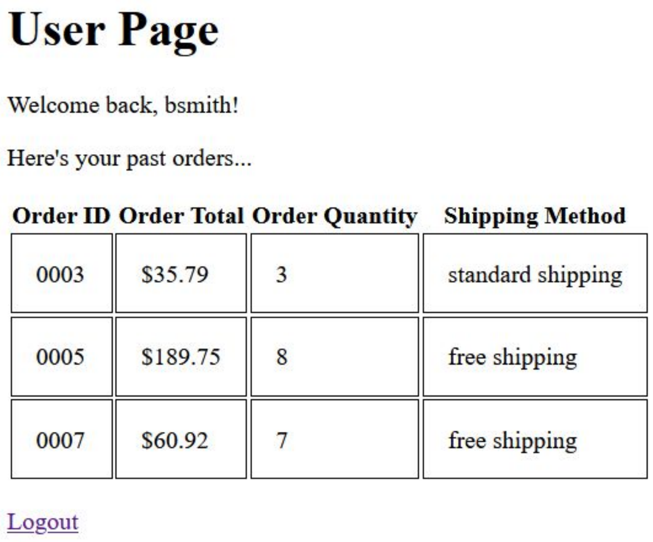
If a user is not logged in and they try to view the User or Admin page by typing in the URL, they should receive an error message and be presented with a link to the login page.

If a User tries to view the Admin page (or vice versa), they should receive an error message and be presented with a link to the login page.

Once a user logs out, they should not be able to view their content page (`user_page.php` or `admin_page.php`) by using the browser's back button or by typing in the URL.

For the `user_page.php` and `admin_page.php`:

The user's login page should display a table of the orders they have placed by linking the currently logged in username with the orders table. An example of what the user page should display is as following:



User Page

Welcome back, bsmith!

Here's your past orders...

Order ID	Order Total	Order Quantity	Shipping Method
0003	$35.79	3	standard shipping
0005	$189.75	8	free shipping
0007	$60.92	7	free shipping

Logout

The admin's page should display a table of all orders, making sure to include the username in this version of the table.

Deliverables:

- login_page.php
- user_page.php
- admin_page.php
- logout_page.php

Upload your PHP files to Canvas. **Include a link to each file on your Pluto account in the submission comment**.

Do not alter any of the username/password combinations added to the database. We must be able to test whether the pages work with valid credentials.

Point Breakdown:
(100 points total)

- login_page.php - 30 points total
  - 10pts - forwards to correct page if already logged in
  - 10pts - creates a session and stores appropriate values on correct login
  - 5pts - displays an error message on incorrect login; repopulates form
  - 5pts - forwards to correct page once correct login
- user_page.php - 25 points total
  - 10pts - displays a table of that user's orders
  - 5pts - displays correctly for Users
  - 5pts - displays error message for a user not logged in
  - 5pts - displays access error for wrong user type access
- admin_page.php - 25 points total
  - 10pts - displays a table for all users' orders
  - 5pts - displays correctly for Admins
  - 5pts - displays error message for a user not logged in
  - 5pts - displays access error for wrong user type access
- logout_page.php
  - 10pts - destroys sessions correctly
- Validation
  - 10pts - all files validate

Hints:

One way to forward the browser to a new page is to send a header in the response that specifies a location. For example, the following PHP code

```
header(“Location: user_page.php”);
```

Would forward the browser to the `user_page.php`