PIC 40A Section 1 - Homework #5 (due Wednesday, May 27, by 6 pm)

Follow these submission instructions carefully:

- submit the .html, .css, .js files, .php, and all supporting files to CCLE by the deadline – there should be 4 PHP files (**index**, **welcome**, **logout**, and **validate**);

- submit a PLAIN TEXT version of your .php files as well;

- submit a file, "User.txt" that provides your PIC username*

- submit the "Honesty.txt" file as described in the syllabus*;

- produce a live webpage* that can be viewed at **www.pic.ucla.edu/˜your_username/HW5/index.php**** and make no changes to the site after the deadline. What you submit on CCLE should be exactly the same files as can be found on your live webpage***;

- follow the established convention: your main HTML file should be **index.php**, which should be located in the folder **HW5** within **public_html**.

*: if your webpage is not live at the precise link given above, where your_username is replaced by your actual user name, you will get 0/10 for display as per the homework grading polices. Your page has to be live and work. No exeptions.

: our servers will not default to **index.php although this is the industry norm. So someone has to enter the full URL.

***: do not modify that live page or any related files in any way after the deadline. As per the syllabus, no matter how small the change may be, any modifications after the deadline will be treated as a case of academic dishonesty.

USER ACCOUNT LOGIN SYSTEM

This exercise requires you to build a login system. The passwords must be hashed! Here is what should happen.

The user can visit the main page **index.php** that allows them to

- enter an email address: it must contain some string of characters before an '@' symbol, followed by another string of characters);

- enter a desired password: it must be at least 6 characters, with letters (upper and lowercase) and digits only;

- either **register** or **log in**.

Remark: you'll need to use that **pattern** attribute to validate the email address and password.

registration

If they choose to register, it should first be verified that they have not already registered with that email address. If they have already registered, they should be told so with a message: "Already registered. Please log in/validate."
If they have not previously registered, they receive an **email validation** and a message appears beneath the form "A validation email has been sent to: [EMAIL ADDRESS]. Please follow the link."

email validation

An email should be sent to their email address with subject "validation" and body: "Validate by clicking here: [**validation_url**]"

log in

If they choose to log in, their email address and password should be verified.
The email address is valid if they have already validated through the email confirmation, otherwise they are told "No such email address. Please register or validate." This error is also displayed if they have not even begun the registration process.
The password must match the email address in order to send them to the **welcome.php** page.

validation_url

This should refer them to the file **validate.php** with a **get** query string storing their email address and a hashed random number from 100 to 50000.

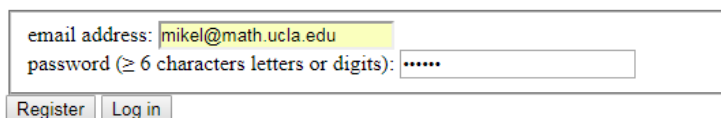The **validate.php** page completes the validation process and tells the user "You are registered!"

The **welcome.php** page should only show content to those who are logged in with a valid email address and password. This page should welcome them, display the email address they logged in with, and list all the email addresses that have been registered! It should also have a **log out** button. When pressed, they are no longer logged in and are taken back to **index.php**.

Do not panic: It is actually not as bad as it sounds. There are just a bunch of small things to get working.
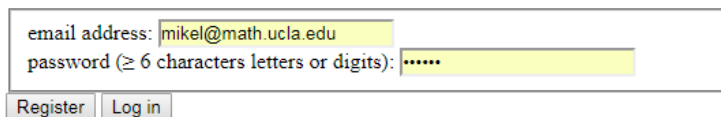
Tips/comments:

- The **get** method is invoked with a query string so as long as you generate a URL with the correct format, your **validate.php** can accept parameters through **get**.

- You'll want two submit buttons on **index.php**. To test which one is pressed, use **isset($_POST['name_of_button'])**.

- To make the **log out** button, you could simply enclose it in an **anchor** that invokes a script **logout.php** that clears the session and redirects to **index.php**.

- You can use either flat files or databases for this. We'll study **sqlite** later on.

- You'll likely want two separate files: one that stores unvalidated users, tracking their email address, hashed password, and a hashed validation token; and another that stores their email address and hashed password after they have been validated. Don't forget to remove them, once validated, from the unvalidated users.

- You'll need to run all of your work through the server on this one.
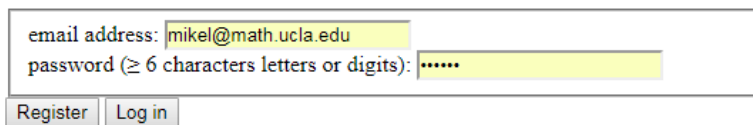
Here are some images:

Figure 1: First visiting the page.

Figure 2: After entering credentials and selecting "Register".

email address: mikel@math.ucla.edu
password (≥ 6 characters letters or digits): ••••••

Register   Log in

Already registered. Please log in.

Figure 3: Trying to register after already registering.

validation
From   mikel@math.ucla.edu
To   mikel@math.ucla.edu
Date   Today 16:14

Validate by clicking here: ███ validate.php?
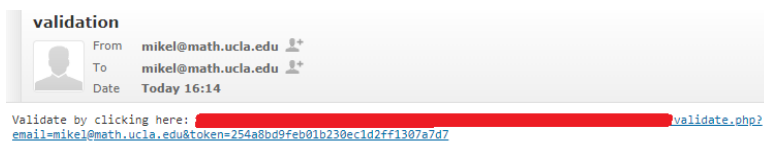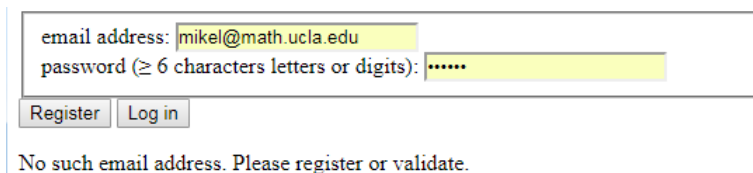email=mikel@math.ucla.edu&token=254a8bd9feb01b230ec1d2ff1307a7d7

Figure 4: The email that is sent. The red parts will be replaced by your own directory structure leading to the php script. Note that the sender will be your PIC email address.

You are registered!

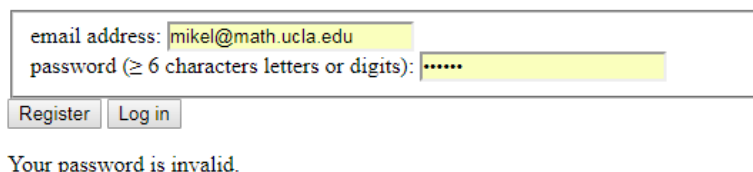Figure 5: Following the link in the email.

email address: mikel@math.ucla.edu
password (≥ 6 characters letters or digits): ••••••

Register   Log in

No such email address. Please register or validate.

Figure 6: If the user logs in but they have never registered or have not yet validated their account.

email address: mikel@math.ucla.edu
password (≥ 6 characters letters or digits): ••••••

Register   Log in

Your password is invalid.

Figure 7: Entered wrong password.

Welcome. Your email address is mikel@math.ucla.edu.

Here is a list of all registered addresses: mikel@math.ucla.edu foobar@gmail.com
log out

Figure 8: After logging in, this is the welcome page.