Working Through Assignment: Login and Registration

Started 04/10/19 1:45

Database results:

For routes, look at users asssmnt and email assgmnt

Html and routes

Html: two pages: reglog.html and success.html

Routes:

* / this is starting route and does little more than render reglog.html the first/blank time

so you always need a home route ('/')

sounds like you'd need a route to handle login logic, so maybe something like ('/login')

and since you'd have different logic for handling registration, you'd likely need a route for that, like ('/registration/')

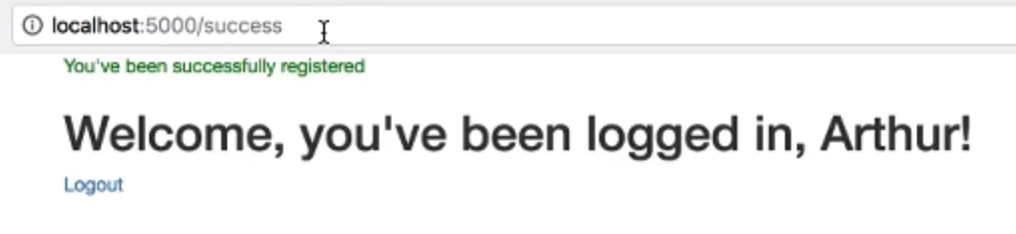
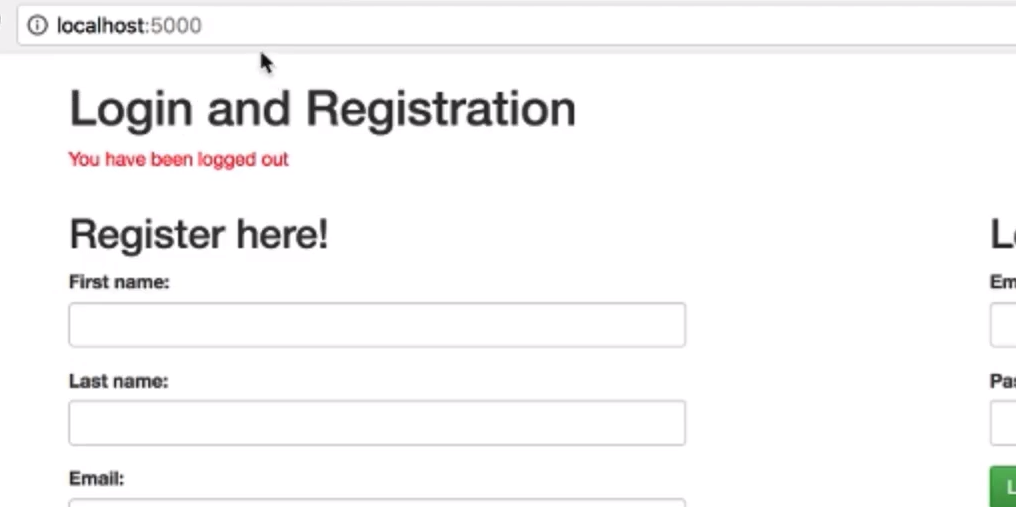
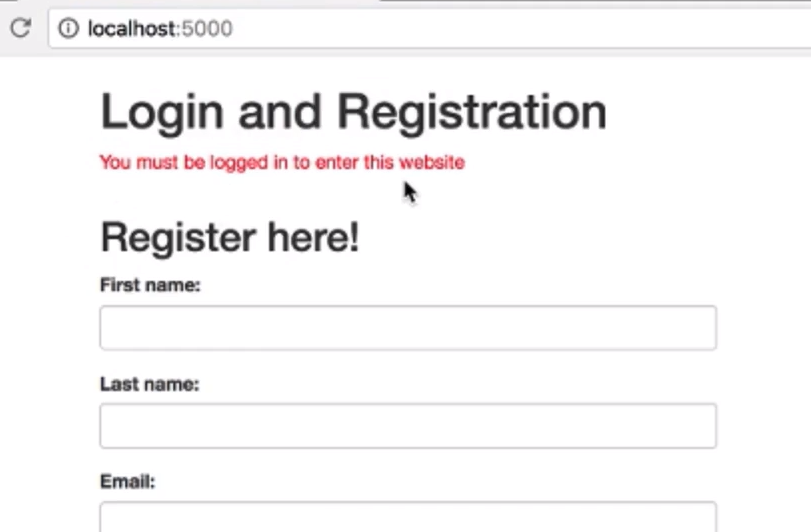
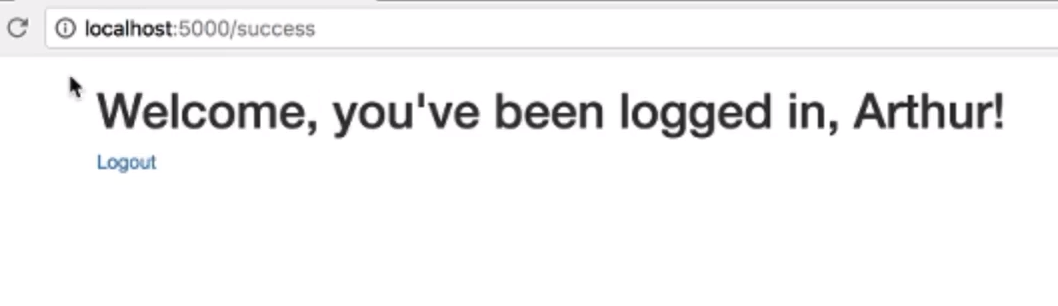
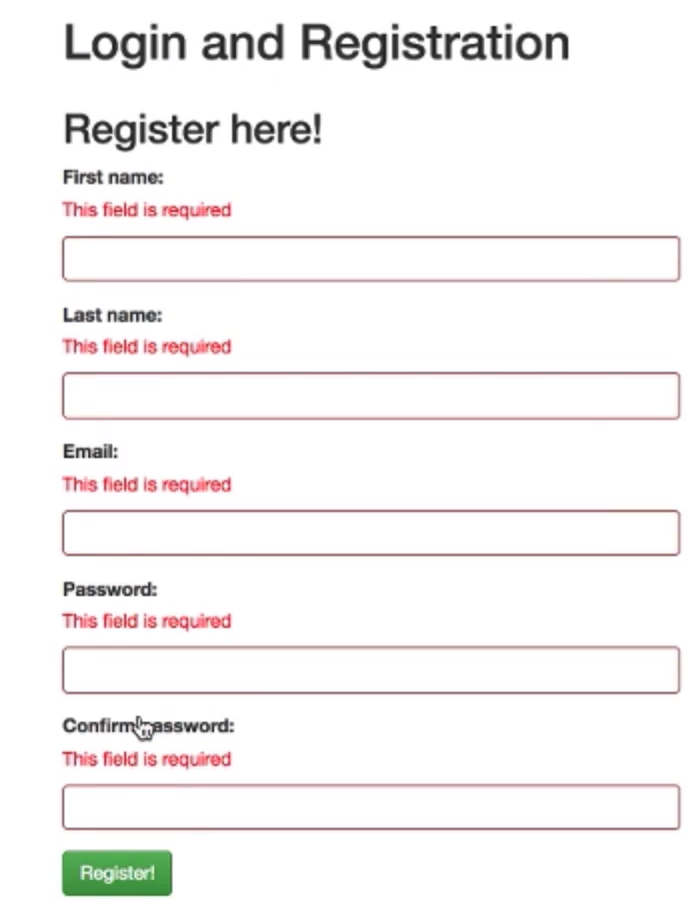
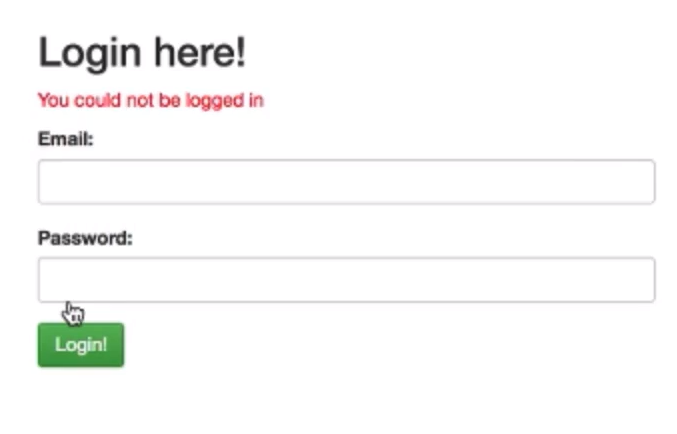
and then maybe a landing page for successful login and/or registraiton, so maybe something like ('/success')

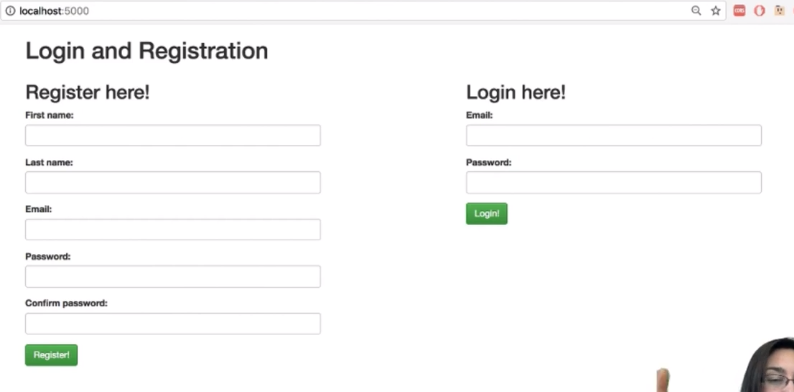
---===Checklist with my comments

* Create a new Flask project
* Create a new MySQL database with a table and the appropriate fields
  + This seems to indicate we only have one table, and if so, just need to name the table..users… and decide on field names. This is provided in the assignment as below:
    - First Name - letters only, at least 2 characters and that it was submitted
      * fname, varchar
    - Last Name - letters only, at least 2 characters and that it was submitted
      * lname vchar
    - Email - valid Email format, does not already exist in the database, and that it was submitted
      * email, varchar
    - Password - at least 8 characters, and that it was submitted
      * Do not store in database
    - Password Confirmation - matches password
      * Do not store in database
    - >>>instead, store hashed password in database named: pw\_hash
* The root route should display a template with the login and registration forms
  + This probably means “/” does nothing more than render an html showing the reg log page…maybe reglog.html
* Validate the registration input
  + This might mean the reglog page sends to the route ‘/createuser’
* If registration is invalid, error messages should be displayed on the index page
  + This means redirecting back to the same page
* If registration is valid, hash the password and save the user in the database, store the user in session, and then redirect to the success page
  + Redirect, or direct?
  + Anyway, this looks like the create route has two possible return statement, and the successful return goes to success.html
* Validate the login input
  + Same page, reglog.html sends to
* If login is invalid, display an error message on the index page
  + Same reglog.html page as used for collecting information
* If login is valid, store the user in session and then redirect to the success page
  + In Amy’s video, same success.html page with exact same display as registration success
* Add a functioning logout button to the success page that clears session
  + See snippets page for how to clear the cookie
* After logging out, ensure you cannot reach the success page
  + Not sure about this, yet, but suspect the route function on the success route (whatever that means) checks the cookie to see if the user is registered…if any user is registered
* NINJA BONUS: Add an additional validation on passwords to have at least 1 number and 1 uppercase letter
* SENSEI BONUS: Add additional input types on the form. Get creative with your validations! (consider including a datepicker, radio buttons, or checkboxes)

----===end Assnmnt checklist

Screenshots and such from the assignment:

* Info from Amy’s video
  + After filling out form, If I have successfully registered…access success route: 5000/success:
  + 
  + Green flash message at the top goes away when page is refreshed
  + Logout button on Success page takes back to route route as below, but this time with a flash message:
  + 
  + Then when she tries the /success route when not logged in, things go back to the same page, but now with a new red flash message:
  + 
  + Now she logs in (already registered) and ends up at the success page, but this time no flash message
  + 
  + Result of submitting empty form…notice flash messages:
  + 
  + She said we could have just put one message at top or all messages at top…doing the messages next to the fields takes extra planning and CSS work
  + Submission with different errors:
  + 
  + She populates the fields with previously attempted values after the rejection
  + Message after bad login data does not give hints as to what the problem is:
  + 



Michael’s video

* If bad log in or registration page, then redirect back to the same page
* When logged in, a session cookie has the user’s user id in it
* Maybe store the first name in session cookie if it is frequently used
* Maybe have a key of logged in which is set to true if logged in