

Login/Register attempt write-up

I attempted to create a login and registers option for users. To start off I kept it simple and tried to have the login as the first page users see when they reach the site and they have to register or login to see the site.

What worked?

- Users could register and they would be added to the database with a hashed password.
- If someone tried to register with a user name already in use they would receive feedback that this was the case and asked to try another or login.
- On registration the user was redirected to the home page.
- The number of characters were restricted and users could not submit unless they were within the limitations.
-

Register Form with user name that already existed entered.

Register

Please enter username and a password to create a account

Sorry, username already taken. Please try another. If you have an account please login

Username:

3-10 characters

Password:

5-12 characters

Register

Already registered? [Login](#)

Issues

When a user tried to login they would receive feedback that the password was incorrect even though I tried for many accounts with the correct password.

Login Form informing the user that the password was incorrect even when it is correct.

Login

Please enter your login details below

Sorry incorrect password!

Username:

Password:

Login

Not registered? [Register](#)

I looked at many examples online and through the slack community that had projects and examples from other students. I did not find one that worked for me. I also contacted tutor support and was sent another example and tried using this method. At this time I knew that even if I got the login to work there was too much I had to add to make a good register login function.

What I learnt

I was so focused on getting the basics in place but needed to consider the time for neassary features such as:

- Confirming password on registration
- The ability for user to reset the password when they forget.
- An account page for users to updates details
- Rukes for passwords such as containg caps, numbers and symbols.

This is the first time I have attempted or even thought about the process of creating an account. It is not a requiremnt for grading and has not yet been part of the course. I think that just looking into various ways devleopers have done this has been a fantastic insight for me. I spent a lot of time wartching videos, reading online tutorials and looking at others works and have found this

valuable to my learning. I simply ran out of time to really look into the bug as I needed to focus on testing and writing this up ready for submission.

HTML for Register page

```
<!--register an account form-->
<main class="container-fluid log-reg-form page-padding-sm">
  <div class="form-block">
    <div>
      <h2 class="headers-position">Register</h2>
      <p class="text-center log-reg-info">Please enter username and a password to create a account</p>
      <form class="log-reg-content" action="{{ url_for('register')}}" method="POST">
        <!--displays feedback messages if username already exists-->
        {% with messages = get_flashed_messages() %} {% if messages %} {% for message in messages %}
        <p>{{ message }}</p>
        {% endfor %} {% endif %} {% endwith %}
        <!--username and password input-->
        <div class="form-group">
          <label for="username">Username:</label>
          <input type="text" class="form-control" id="username" name="username" placeholder="3-10 characters" minlength="3" maxlength="10" required>
        </div>
        <div class="form-group">
          <label for="password">Password:</label>
          <input type="password" class="form-control" name="password" id="password" minlength="5" maxlength="12" placeholder="5-12 characters" required>
        </div>
        <!--Submit button and link to login-->
        <button type="submit" class="btn form-btn green-button">Register</button>
        <p class="log-reg-link">Already registered? <a href="{{ url_for('login')}}"> Login</a></p>
      </form>
    </div>
  </div>
</main>
```

HTML for Login page

```
23 # route to login page
24 @app.route('/')
25 @app.route('/login', methods=['POST', 'GET'])
26 def login():
27     """Function for handling the logging in of users"""
28     if 'logged_in' in session: # Check is already logged in
29         return redirect(url_for('index'))
30
31     if request.method == 'POST':
32         user = mongo.db.users
33         logged_in_user = user.find_one({
34             'username': request.form['username'].title()})
35
36         if logged_in_user:
37             if check_password_hash(logged_in_user['password'],
38                                   request.form['password']):
39                 session['username'] = request.form['username']
40                 session['logged_in'] = True
41                 return redirect(url_for('index'))
42
43         flash('Sorry incorrect password!')
44         return redirect(url_for('login'))
45     return render_template('login.html')
46
47 @app.route('/register', methods=['GET', 'POST'])
48 def register():
49     """Function for handling the registration of users"""
50     if 'logged_in' in session: # Check is user already logged in
51         return redirect(url_for('index'))
52
53     if request.method == 'POST':
54
55         user = mongo.db.users
56         dup_user = user.find_one({'username': request.form['username'].title()})
57
58         if dup_user is None:
59             hash_pass = generate_password_hash('password')
60             user.insert_one({'username': request.form['username'].title(),
61                             'password': hash_pass})
62             session['username'] = request.form['username']
63             session['logged_in'] = True
64             return redirect(url_for('index'))
65
66         flash('Sorry, username already taken. Please try another. If you have an account please login')
67         return redirect(url_for('register'))
68     return render_template('register.html')
69
70
```

25:46 Python Space

Python code

This also had a simple logout @app.route that ended the session, and redirected the user to the login page. This was added to the main navigation and did work as expected after a user registered.

Session was imported from flask and werkzeug was installed and added to the app.py file:

```
from werkzeug.security import generate_password_hash, check_password_hash
```

```
23 # route to login page
24 @app.route('/')
25 @app.route('/login', methods=['POST', 'GET'])
26 def login():
27     """Function for handling the logging in of users"""
28     if 'logged_in' in session: # Check is already logged in
29         return redirect(url_for('index'))
30
31     if request.method == 'POST':
32         user = mongo.db.users
33         logged_in_user = user.find_one({
34             'username': request.form['username'].title()})
35
36         if logged_in_user:
37             if check_password_hash(logged_in_user['password'],
38                                   request.form['password']):
39                 session['username'] = request.form['username']
40                 session['logged_in'] = True
41                 return redirect(url_for('index'))
42
43         flash('Sorry incorrect password!')
44         return redirect(url_for('login'))
45     return render_template('login.html')
46
47 @app.route('/register', methods=['GET', 'POST'])
48 def register():
49     """Function for handling the registration of users"""
50     if 'logged_in' in session: # Check is user already logged in
51         return redirect(url_for('index'))
52
53     if request.method == 'POST':
54
55         user = mongo.db.users
56         dup_user = user.find_one({'username': request.form['username'].title()})
57
58         if dup_user is None:
59             hash_pass = generate_password_hash('password')
60             user.insert_one({'username': request.form['username'].title(),
61                             'password': hash_pass})
62             session['username'] = request.form['username']
63             session['logged_in'] = True
64             return redirect(url_for('index'))
65
66         flash('Sorry, username already taken. Please try another. If you have an account please login')
67         return redirect(url_for('register'))
68     return render_template('register.html')
69
--
```

25:46 Python Spaces

Users added to MondgoDB with hashed password

+ Create Database

NAMESPACES

knowledge_is_power

fundraisers

users

myTestDB

task_manager

knowledge_is_power.users

COLLECTION SIZE: 1.91KB TOTAL DOCUMENTS: 13 INDEXES TOTAL SIZE: 36KB

Find Indexes Aggregation Full Text Search

INSERT DOCUMENT

FILTER

("filter":"example")

Find Reset

QUERY RESULTS 1-13 OF 13

_id: ObjectId("5e268cc4fca5bd8a64bb5877")

username: "Billy"

password: "pbkdf2:sha256:150000\$13c6cp6a5dc4487e9eec4941e79b2fa09f2a0c7732f49f44f..."

_id: ObjectId("5e268d96555f0662d2581805")

username: "Dave"

password: "pbkdf2:sha256:150000\$0ur7kFZJ6s2c564d2d237159e2be673a24ee59470f96b378f..."

_id: ObjectId("5e2769f5034c789e3569f7ea")

username: "Sophie"

password: "pbkdf2:sha256:150000\$TW1oz3ML\$9c6c6f57b1ad7592ff09a78c4dccf1126f4a4873..."

_id: ObjectId("5e1b27bf428ba3db848022d3")

username: "Sam"

password: "pbkdf2:sha256:150000\$GKTC9yRt5e23112f455478fd701b453c0d0691ee690f6e55..."

_id: ObjectId("5e21d16780a66ac0e614974c")

username: "Angie"

password: "pbkdf2:sha256:150000\$58dn1oEC\$4134u945385c368fcfc7b0ee38f499897bb986d..."