Lab 5: Sessions and Authentication

In this lab you will use server-side session management features to store user authentication information. You will build log-in and log-out functionality that can be reused for any number of future server-side applications. You will also implement password hashing and salting, an additional security feature to further hack-proof your applications.

Objectives & Outcomes

Successful completion of this activity will show that you can:

- Implement session management in a server-side application.
- Implement password hashing and salting.
- Implement user log-in and log-out in a server-side application.

Level of Effort

This activity should take approximately 4h to complete. It will require:

- 30m Research
- 5m Prep & Delivery
- 3h Work

If you find that this activity takes you significantly less or more time than this estimate, please contact me for guidance.

Instructions

This lab is very straightforward: implement authentication in both PHP and CFML. You should have:

- A log-in form that prompts for an email address and password, using the POST method.
- A Controller that validates the data (both should exist, the email address should look like an email address, and the password should be at least 8 characters).
- A Service or Model class that fetches the user profile information given the email address and password.
- User records in the table should be salted and hashed. You may use a dedicated salt column or the email address for the salt.
- User profile information stored in the Session scope by the Controller, assuming proper credentials.
- A log-out Controller that clears the user information from the Session and recycles the session if necessary.
- Views for your log-in form, log-out confirmation, and some sort of user home page that shows that the user logged in successfully. (This can be as simple as a dump of the Session scope.)

These must be done in **both** PHP and CFML.

Above and Beyond

Authentication is a pretty simple topic, so there isn't much opportunity to go above and beyond here. You could spend time polishing your views to integrate with your previous User Management application.

Git Commits & Video Reflection

The same rules about Git Commits and Reflection Videos apply for this Lab as for previous labs. Review the Lab 1 documentation to ensure you meet all of the required points. Your required questions are mostly the same, with small differences:

1. What didn't go as well as you had hoped? Where did you run into problems with the lab?
2. What went better or was easier than you expected? What went right?
3. **In your own words, explain *and show* how salting and hashing passwords helps to improve security.**
4. How do you think what you did in this lab is going to help you in your future school work, career, or life?

Where should you be?

Basic PHP and CFML syntax should be mostly no-brainers by now, though you may occasionally put a $ where a # should be or vice-versa. Working with classes and objects in both languages should feel comfortable, as well.

Deliverables

Make sure your source code, assets, and reflection video URL are committed to the repository. Push your changes to the server before the start of the next lecture.