

### Project 3

Some notes before explaining the workflow of project 3:

- The bookstore folder contains the Eclipse project and the database file.
- An additional .jar file has been included in an attempt to solve the SQL injection portion of the project, but it serves no purpose for this project.
- This project is based on my previous project 2 materials (with some extra tweaks). So the prepare statements used for the queries were not changed and are currently used to satisfy the SQL injection protection portion of the project.
- However, my project 2 was also unfinished, so I had to work on it before starting on this project. The only changes I have made was to search users, and corrected the issue with displaying the full address string. The statistics page have been left alone since the Professor assured us that only the current implementations would be graded

When the project is loaded in Eclipse, start it up through the Login.jsp page where the user will be greeted with a typical login page and an option to register if the user has not created an account. For this example, the following credentials are provided:

user: pat, password: asdfghjkl1

user: kat, password: applepie1

where pat is an administrator and kat is a normal user.

If the user selects the "Register" button, they will be greeted with a registration page to enter their credentials. If they try to submit new user information that matches an existing username on the database, an error will prompt saying that the username is already taken. Additionally, if the user enters a password that does not meet the criteria (at least 8 characters in length with at least 1 letter and 1 digit), they will also receive an error prompt.

If the user does successfully register, the password is hashed using the MD5 algorithm before being stored on the database. When the user goes to login with their credentials, the password they enter also gets hashed and then compared to the hashed password on the database to verify the account.

When a user is not logged in and attempting to view pages that requires a logged in user, they will be able to view the pages, but not be able to make any changes. For instance, using "http://localhost:8080/bookstore/ShowBooks.jsp", if the user is not logged in, they notice that they cannot place any orders, but can view the currently available books at their leisure. If they need to make an order, they must select the "Login" button where they will be taken to the Login.jsp page with a new URL ("http://localhost:8080/bookstore/Login.jsp?returnFrom=ShowBooks.jsp"). Once the user logs

into their account, they will be taken back to the ShowBooks.jsp page so they can make an order. In other pages like ShoppingCart.jsp and Checkout.jsp, the user will not be able to view or any orders made unless they login.

### **Comments**

Due to time constraints, I was unable to fulfill some aspects of the project. For example, I was unable to make a guest user (a user that has not logged in) be able to make orders, but be directed to the Login.jsp page to login. Then be able to return to with the order they just tried to make.

Another issue I had was that I could not apply the hashed password to the ModifyUsers.jsp page for the administrators. Since my logic was that the inputted password be hashed, compared to the password on the database, and then either updated or left alone if they matched or not. Since a hashed password cannot be "unhashed", this made my endeavor much more difficult. Thus, I was unable to figure this out and left it alone.

This project is relatively easy compared to the previous project and I was able to finish well before the due date, but I was unable to fulfill some smaller project requirements. I also was unable to understand the criteria for the project and had to rely on Piazza questions and answers to understand what was requested.

Please make the project easier to understand, not only for myself, but for the TA as well.