

# Final Project

## Overview

At this point, all of your webpages should exist in the form of servlets instead of static html pages. Those same servlets should also receive data from various forms, perform certain calculations on that data, and output the data “where it goes” on the webpage instead of a blank webpage. They should also implement sessions to handle login/logout and keep track of what form data the user has submitted. Additionally, that data should be persistent through server reboots by using a MySQL database.

For the final project, you will add the “icing on the cake” so to speak. You will create and populate a bean when the user logs in. You will also convert a provided html file to a jsp file and use it to display the bean. Additionally, you will use ajax on your registration page to detect whether or not a typed in username is already in use.

**75 percent of the grade is based on the requirements of the previous assignments.** If you have not completed all of the requirements of the previous assignments, you are encouraged to do so before beginning the final project, as the final project directly builds upon them. As always, you may change your mind about the final project and choose a different option, but the previous content for the other option will need to be implemented before beginning the final project.

## Final Project Requirements

For the final project, there are a few new requirements, but not much has changed from the previous assignments. Since the grade weight is much higher, this is an opportunity to greatly increase your grade if you missed a lot of points on assignment 3 or 4. For the final project, the requirements of assignment 4 still apply.

Although an option on some previous assignments, please **do not use html 5 “required” attributes** for the final project. I must be able to tell that your server side validation is working.

### 1. User account registration page

In addition to the assignment 4 requirements, this page in the final project contains the following new requirements:

When a user attempts to create an account and they click off of the username field (an onmouseout javascript event), Javascript should use ajax to send a request to a new servlet or jsp file. The new servlet or jsp file should check the database to determine whether or not the username is already in use. The servlet or jsp file should then return its answer to the javascript. If the username is already in use, the javascript should modify the page to say “username already in use.” If the username is not in use,

the javascript should modify the page to say “username available.” The servlet which creates accounts (the page itself) should also be modified to check whether or not the username is in use as part of its normal validation procedure. If the username is in use and the form is submitted, the page should not pass validation, the account should not be created, and the page should display an error message stating what happened. This error message may be in the same location as the error displayed by the javascript.

## **2. Home/Login page**

In addition to the assignment 4 requirements, this page in the final project contains the following new requirements:

When the user logs in, a bean should be created by the servlet. This bean should be filled with the user's data that they provided when they created the account. Then the bean should be placed on the session.

An html page has also been provided to you for use in your final project. It is in moodle. Please convert this webpage to a jsp file and include it in your application. This jsp file should receive the bean placed in the session and display the data. I do not care where on the page the data from the bean is displayed or how it looks. You do not have to alter the style of this page to match the rest of your application. It does not need links. Please use the same filename with a jsp extension. You may use EL or JSTL if you choose. **Do not use scriptlets. No credit will be given if you use scriptlets.** This is because it bypasses the point of mvc. Note that scriptlets are not the same as servlets.

The purpose of the above is to simulate being handed a “view” in the real world by a web developer. The idea is to gain practice filling the view with data without worrying about the “lowly” style issues best left to lower paid web developers.

## **3. Content Page 1**

There are no new requirements for content page 1.

## **4. Content page 2.**

There are no new requirements for content page 2

## **5. Content page 3. (if online marketplace option is chosen)**

There are no new requirements for content page 3

## **Extra Credit**

You may earn extra credit worth 5 points on this assignment. If you wish to earn this extra credit, you have two choices no matter which project option you chose. No matter which option you choose, you must **fill out the short word template in moodle and submit a copy with your assignment** as a separate file. The extra credit options are listed below.

### **Option 1- Use a web service**

If you choose this option, you must locate and use a webservice to do something useful in your application. If you are unsure about what counts as “useful” in your application please ask me at least one week prior to the deadline. As an example, you could use the yahoo finance api to maintain live real life stock prices (if you are using the stock exchange option). Here is an example of the api

and a query in the url:

<http://finance.yahoo.com/webservice/v1/symbols/COALINDIA.NS/quote?format=json&view=detail>

### **Option 2- Use more ajax**

If you do not wish to find and use a webservice to do something useful, you may alternatively implement more ajax on your website. If you are doing the stock exchange option, when users attempt to buy or sell stocks, ajax should be used to perform the trade. The page should be updated with the appropriate messages and content (see previous assignments for details) without refreshing the page. If you are doing the online marketplace option, ajax should be used to add items to the user's cart when the user clicks the "add to cart" buttons. The page should not be refreshed when items are added.

### **Independent Projects**

If you proposed an independent project and it was approved, in this assignment, you must adhere to the same requirements for the registration page and the login page as the other assignments. This means you must include the jsp file. The old assignment 4 requirements still apply. If you have questions about your project, please contact me at least one week before the due date. For the 5 points of extra credit, you may complete "Option 1" detailed above. If you wish to do something similar to "Option 2," Please contact me at least one week before the due date, explain what you are going to do, and get approval.

### **Additional Requirements for All Assignments**

The site should use css to implement some sort of simple color scheme and be reasonably aesthetically pleasing. (Basically it should not look 'tacky').

All pages should be linked to all the other pages. Links should be in a consistent position. They should not move or break from page to page.

### **Submission Instructions**

When you submit this assignment, in addition to the submission url and zip file, you must submit an sql script file which sets up your database and tables on my system. This is because you cannot "upload" a database into moodle. This script is a file with an sql extension and several sql statements which create a database, create a table or several tables, create a user, and assign permissions to that user. I have created a template for you to use and uploaded that template into moodle. You should be able to fill out the template with the necessary information for your database. **IMPORTANT: Please verify your sql script works before turning in your assignment. If your database does not setup properly, the assignment will receive a 0.** You can test this by deleting your existing database (preferably after backing it up) and running the statements contained in your .sql file. If your app works properly after this, your sql script is good to go.

**If you are using netbeans**, please make sure that you correctly **EXPORT TO ZIP** within netbeans as

shown in class. This means, within netbeans, click “file” at the top left, then navigate to “Export Project” and choose “To Zip...” This will open a dialog box. Select your project and click the “Export” button. Do not simply zip the directory itself with windows or your project will not load into my netbeans installation correctly. **If it does not load into my netbeans installation correctly, you will get a 0 on your assignment.** To test whether or not you are doing it correctly, please backup your project, export it from netbeans, delete it from netbeans, and import your exported copy. As with the last assignment, you do not have to adhere to any naming convention, but if you would like to, you may name your project NetbeansLastname where Lastname is your last name. Please upload this zip file into moodle.

**If you are using eclipse**, the instructions have also changed. The zip files do not load properly into netbeans so I will be grading your applications with a separate eclipse installation. Eclipse has the ability to easily export war files with the source code. **Please export a war file of your project** by right clicking your project within eclipse and navigating to the option called “Export.” Choose “war file” and a dialog will appear. Check the box for “export source files” and click finish. Save your file somewhere. Rename your file “EELastname” where Lastname is your last name. Please upload this file into moodle. Please adhere to this naming scheme so I know to use eclipse to load your application. Do not simply zip the directory itself with windows or your project will not load into my eclipse installation correctly. **If it does not load into my eclipse installation correctly, you will get a 0 on your assignment.** To test whether or not you are doing it correctly, please backup your project, export it from eclipse, delete it from eclipse, and import your exported copy.

**For both netbeans and eclipse**, provide a url for your application's home page or login page in the comments section in moodle. Unlike 2300, this will be a local url. It will probably look like one of the following

<http://127.0.0.1/yourpackage/yourervlet>

or

<http://localhost/yourpackage/yourervlet>

but definitely not

<C://users/name/workspace/somefile.html>

Please do not attempt to use notepad++ or dreamweaver for this assignment. They both work great for simple web development but do not support dynamic web development at all.

## Grading Rubric

10 points -All site content present and site links working and in consistent position. Css styles

implemented. Site has color scheme and has a good look and feel. All pages are generated from servlets instead of existing as static html files. Servlets do not crash when some form data is not provided to them, and instead display the appropriate messages.

5 points- Server side account validation and creation works, even if users are not saved persistently.

Login page works correctly. Redirection when users are not logged in works correctly.

10 points- All additional assignment 3 functionality working. This includes purchasing/selling of stocks for stock sites and carts for marketplace sites.

10 points- User accounts correctly being stored in database when account is created. Login works after server restart. Correct user balance and user account creation data persist through server reboots. This information is displayed on the correct page when the user logs in after a server reboot.

40 points- Other assignment 4 requirements met. This includes, among other things, correct displaying of persistent stock holdings for project option 1 or persistent order history for project option 2. This information must persist through server reboots and be stored in a database.

5 points- Server correctly checks whether or not username is in use when attempting to create user account and outputs appropriate messages.

10 points- Bean is created and populated when the user logs in and the new jsp page shows the data in the bean.

10 points- Ajax and javascript working correctly in registration page.

5 points- Extra credit option completed

Total 105 points

### **Submission Checklist (NOTE: This has changed in the final project)**

Don't forget: For the final project you need to turn in these 4 things:

1. Submission link
2. Sql script
3. The correctly exported zip file.
4. A filled out extra credit template explaining what you did for extra credit (if you did the extra credit)