

PROG2240: Java Web Tech

Assignment 3

Winter 2022

Due Date: 9th April, 2022

Background:

In this assignment, you will continue to use the MVC pattern using servlets and JSPs. You will use the servlets with *action* pattern for Cart servlet, JSP EL and JSTL to develop the **XXYY** Computer Programming Club's **Loan Cart** Web application. You will also use the various scope (i.e., page, request, session, and application) of EL when accessing attributes.

Additional Requirements:

- Explicitly specify the *scope* when accessing attributes using EL (e.g., *sessionScope*).
 - Do **not** use `<c:import>` tag when `<jsp:include>` is adequate.
 - Do **not** use `sendRedirect()` method when `forward()` method is appropriate.
 - Do **not** use scripting elements (e.g., JSP Scriptlet, JSP expression) when JSP EL is available.
-

Tasks

1. Setup XXYYClub NetBeans project

a. Add Java classes

- Download **A3Files.zip** and copy these two Java classes to the existing **club.business** package.
 - [ELoan.java](#) and [ECart.java](#)
 - Study all the new code provided
 - Use the methods provided as part of your solution (see next steps).
- Important: **Do not modify any of these classes**. Also, **do not** use *BookIO class* for A3.

b. Ensure that the JSTL library is available for this application

2. Modify the Banner JSP page

- Add two new hyperlinks to the [XXYYBanner.jsp](#).
 - [eLoan](#) (linked to [XXYYLoan](#) servlet -- see Step 3a)
 - [eCart](#) (linked to [XXYYCart](#) servlet -- see Step 4a)
 - For the screen layout of these hyperlinks, refer to the sample screenshot in Step 3.
-

3. Write the eLoan servlet and JSP

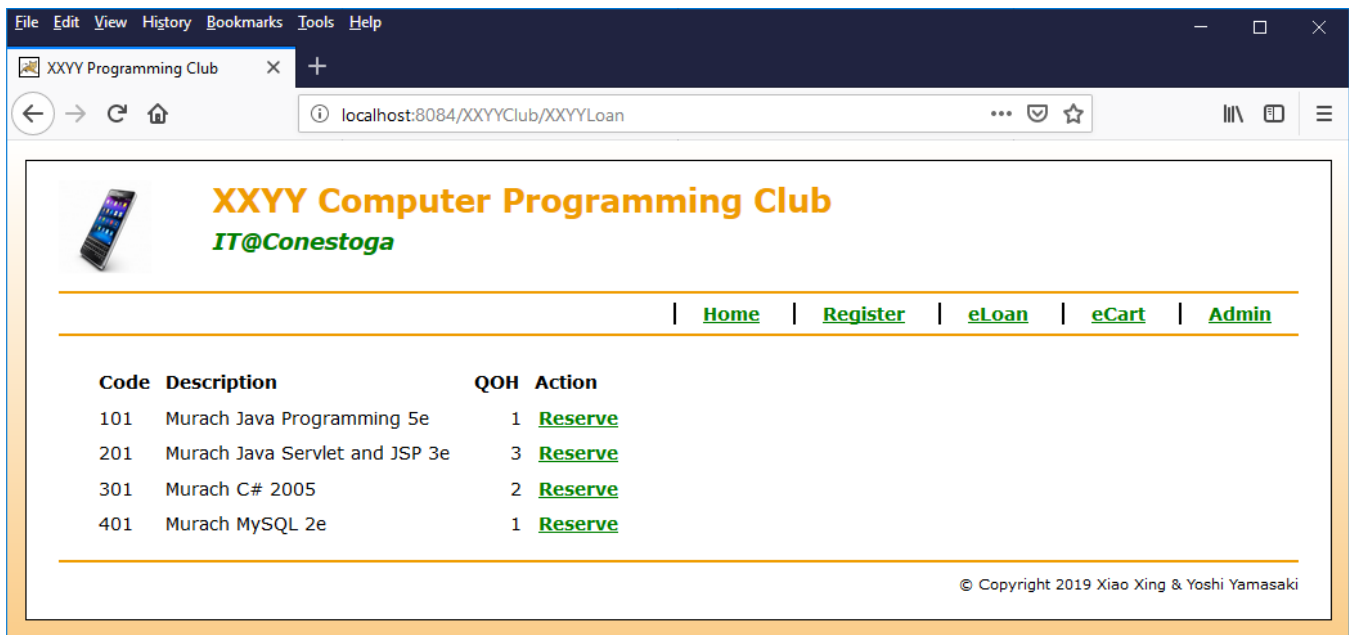
a. Add an eLoan servlet

- Under the Java source (i.e., "[Source Packages](#)") directory, add a new package: **club.cart**
- Add a servlet named `XXYYLoanServlet.java`
 - Set URL pattern to [/XXYYLoan](#)
- Within the **processRequest()** generated method:
 - Create a new **servlet context** object (i.e., [application](#) object)
 - Using the context parameter prepare the full path name of `books.txt` file.
 - Maintain a **loanitems** object to store the ArrayList of Book objects available on loan.
 - *Hints:*
 - Create/save the **loanitems** object in the **servlet context** object
 - If **loanitems** stored in the **servlet context** object is null, call the **loadItems(string)** method of **ELoan** class. This will populate **loanitems** object with the list of books from the **books.txt** file.
 - Forward the control to [XXYYELoan.jsp](#) page.

b. Add an eLoan JSP page

- Add a JSP page named [XXYYELoan.jsp](#).
- Add the banner and footer pages.
- Using [JSTL](#) and [EL](#) (**not** scripting elements), add the code to match the design layout shown below.
- The **eLoan JSP** page shows a list of **loanitems** with the quantity-on-hand (QOH)
- Each time you click the [Reserve](#) hyperlink, call the [XXYYCart](#) servlet with two parameters
 - **action**: set to **"reserve"**
 - **code**: set to the **code** selected (e.g., **101**)

- Note: The **XXYYCart** servlet must require these two parameters to add items to the cart



Note: If the **Reserve** link is selected, call the **XXYYCart** servlet (see next step). When items are reserved (i.e., added to the loan cart), the QOH of that loan item is reduced by 1. If the QOH becomes zero, the **Reserve** link becomes "N/A" similar to the screen shot shown below (see Step 6, QOH for **code 101**).

4. Write the eCart servlet and JSP

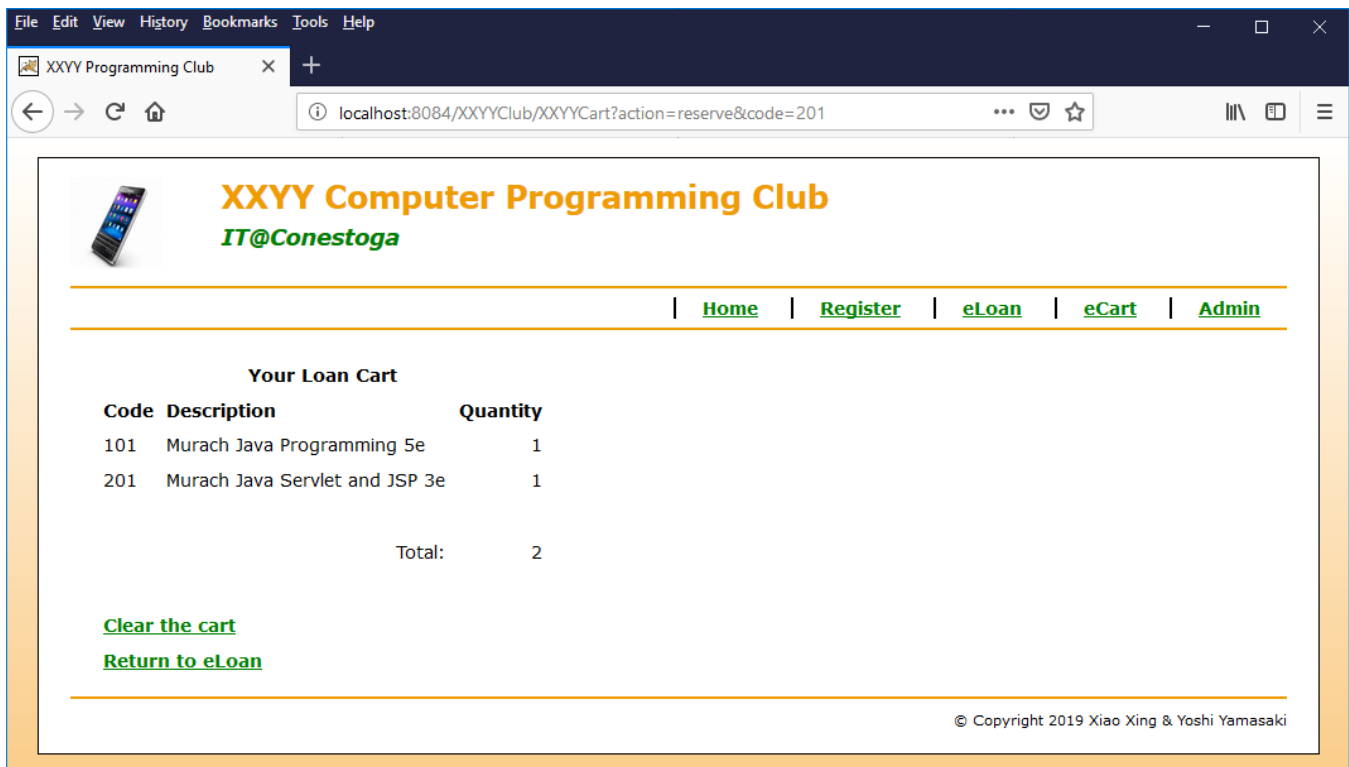
a. Add the eCart servlet

- Under the **club.cart** package, add a servlet named **XXYYCartServlet.java**
 - Set URL pattern to **/XXYYCart**
- Within the **processRequest()** generated method:
 - Access the **loanitems** object stored in the servlet context.
 - Create/save a loan **cart** object (instantiated from **ECart** class) in the **session** object
 - If action is "**reserve**":
 - Using the "**code**" parameter, get details of the loan item (i.e., Book object) by calling the **findItem(ArrayList, String)** method of **ELoan** class.
 - Call the **addItem(Book)** method of the loan **cart** object

- Reduce the loan item's QOH by 1
- *Hint:* Call the **subtractFromQOH**(ArrayList, String, int) method of the **ELoan** class.
- Forward the control to **XXYYECart.jsp** page

b. Add the eCart JSP page

- Add a JSP page named **XXYYECart.jsp**.
- This page will display all the loan items added to the **cart**.
- Add the banner and footer pages.
- Using **JSTL** and **EL** (not scripting elements), add the code to match the design layout including the title shown below.
 - Show **total quantity** in the cart (i.e., reserved).
 - *Hint:* Use the **<c:set>** JSTL tag to declare a variable.
 - **Quantity** column should be aligned to the right.
- Sample screen shot when **Reserve** link is selected for both **code 101** and **code 201**:



5. Write the ClearCart servlet

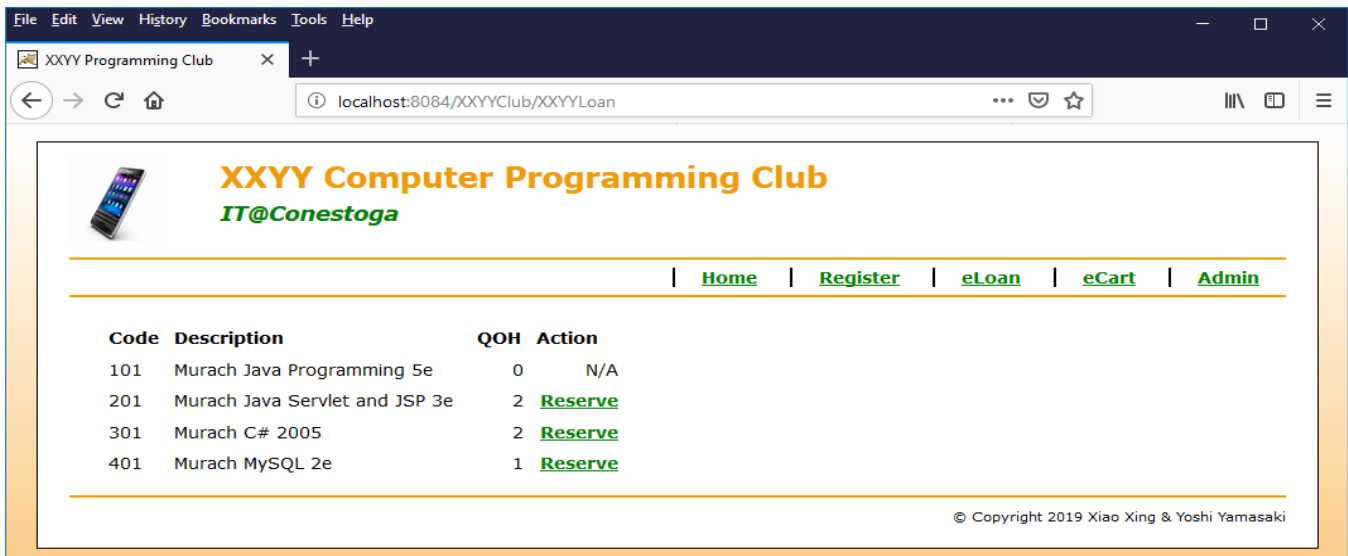
- Under the **club.cart** package, add a servlet named **XXYYClearCartServlet.java**
 - Set URL pattern to **/XXYYClearCart**
- Within the **processRequest()** generated method:
 - Create/save a loan **cart** object (instantiated from **ECard** class) in the **session** object, *as appropriate*.
 - Use the **getItems()** method of the **cart** object to access the list of cart items
 - Add back the quantity reserved from each cart item to the loan items QOH.
 - Hint:* Call the **addToQOH(String, int)** method in the **ELoan** class.
 - Invalidate the **session** object
 - Note: This will delete the current session ID and start with an empty cart
 - Forward the control to **XXYYECart.jsp**

6. Continue to Test the Loan Cart Web Application

Important Note: If you encounter any errors/issues during testing, you must modify your code to resolve all the errors/issues.

a. After adding code 101 and code 201 to the cart, click the [Return to eLoan](#) link

- [Murach Java Programming 5e](#) book has been reserved and QOH is **zero**. Action column is shown as "N/A".
- [Murach Java Servlet and JSP](#) book has been reserved and QOH is 2.



XXYY Computer Programming Club
IT@Conestoga

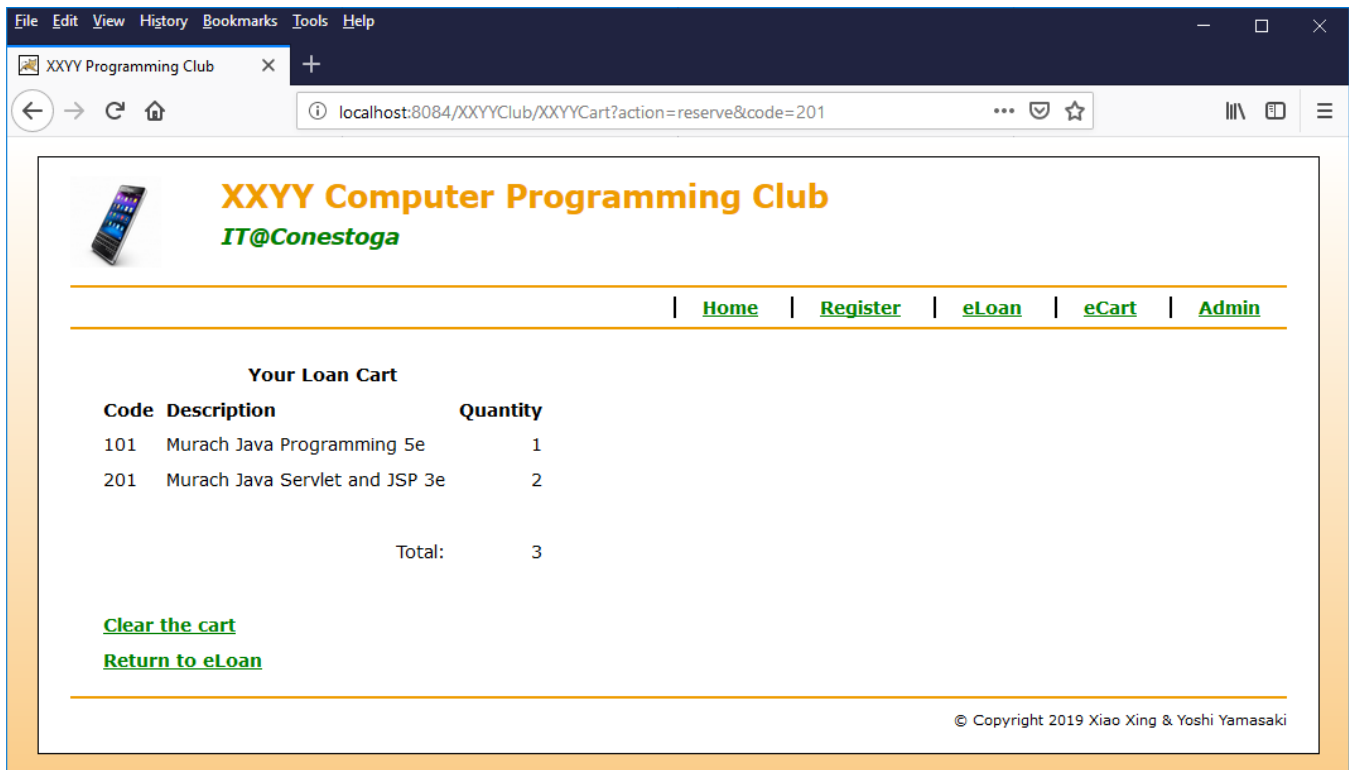
Navigation: [Home](#) | [Register](#) | [eLoan](#) | [eCart](#) | [Admin](#)

Code	Description	QOH	Action
101	Murach Java Programming 5e	0	N/A
201	Murach Java Servlet and JSP 3e	2	Reserve
301	Murach C# 2005	2	Reserve
401	Murach MySQL 2e	1	Reserve

© Copyright 2019 Xiao Xing & Yoshi Yamasaki

b. Click [Reserve](#) link for **code 201** again.

- Result: The **Cart** is displayed with a total quantity of 3 books in the cart.



c. Click the [Clear the Cart](#) link

- Result: All cart items are removed from the **cart** object. The QOH of the loan items list must be updated accordingly.

d. Click the [Return to eLoan](#) link

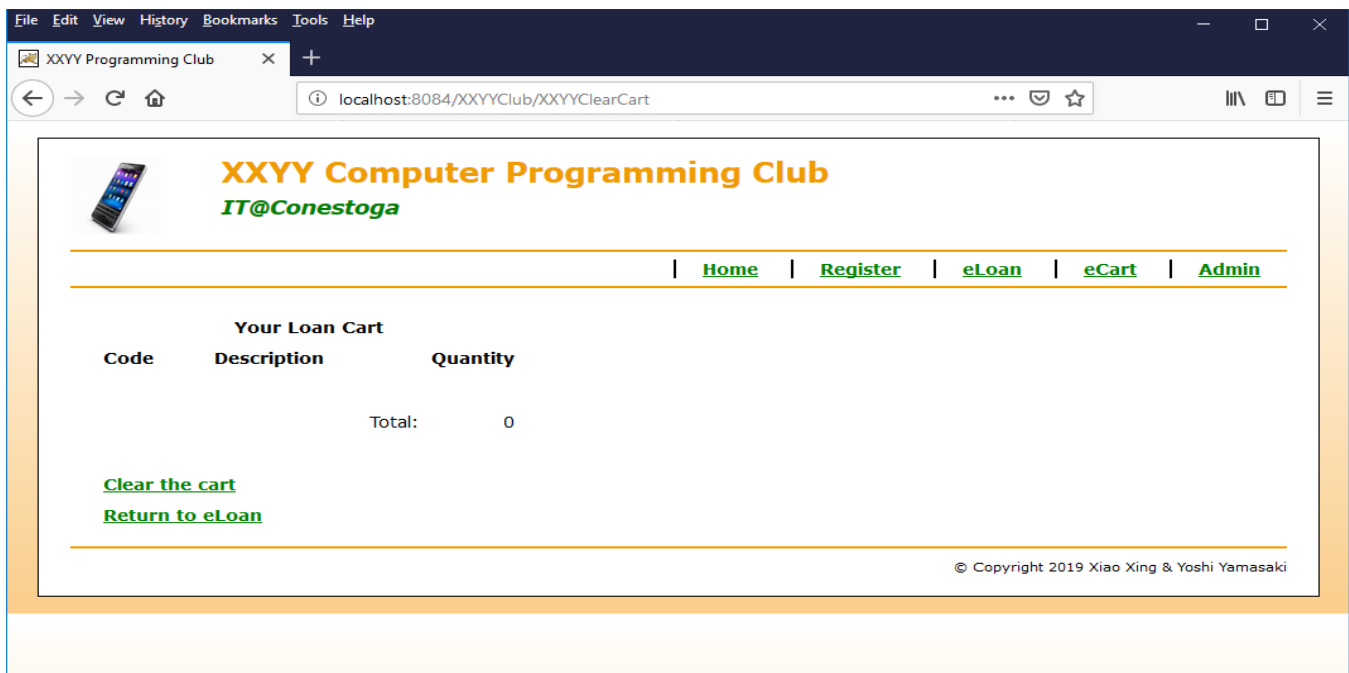
- Result: All items are now available to be reserved with QOH shown in sample screen shot of Task 3.

e. Testing an Empty Cart

- From the **Banner JSP** page, click on the [eCart](#) link.
 - Result: Empty Cart (see sample screenshot below)
- From the **eCart** page, click on [Clear the Cart](#) link.
 - Result: Empty Cart (see sample screenshot below)

f. Build and exercise your own negative test cases

- E.g., On the Web browser, directly enter a URL to execute Cart servlet without the **action** parameter.



7. Testing Multiple Sessions

While the first **session** (i.e., Web browser) is still open, start another Web browser session.

- Start | All Programs | Google Chrome (or FireFox)
- Run the **eLoan** page.
 - E.g., <http://localhost:8084/XXYYClub/XXYYLoan>
- In the second session, click **Reserve** link
- **Result:**
 - The QOH must show the loan items that are still available. Loan items reserved by the first (previous) session cannot be reserved by the second (new) session.
 - The cart items in the first session are different from the cart items in the second session. This demonstrates the use of the **session** implicit object.

Tip: To reset all the QOH to the original quantities, **restart** the Tomcat server.

8. Prepare screenshots of your Web Application

- Create a new **A3_screens.docx**.
- Add at least **5 screenshots** similar to the above screenshots (with the **title** displayed).
 - Each screenshot should have an *appropriate title*.

Assignment Submissions

Important You must submit your code and screenshots (zip file) and upload your solution (XXYYClub.zip).