CPRG352 - Web Application Programming

Fall 2021

Topic: State Management

YOU MAY USE JUST SERVLETS, OR A COMBINATION OF JSPs AND SERVLETS, TO IMPLEMENT ALL THE FUNCTIONALITY IN THE APPLICATIONS BELOW

Problem 1: Storing State using a *HttpSession*

Create a web application called **WebCounterSession** that lets a user increment a number, or reset the count to zero. *(The functionality for this application is the same as for that of the WebCounter you implemented using a hidden form field in the previous lab.)*

When first run the application should show the following form:

Clicking on the **Increment** button adds one to the count value each time it is clicked.

Example: the **Increment** button was clicked three times by the user

Clicking on the **Reset** button resets the count to zero.

Example: User clicked on the **Reset** button

Application Requirement:

The count value must be stored as an attribute in a **HttpSession** available in the servlet. The **Increment** and **Reset** buttons are form submit buttons that submit the form to a servlet. The servlet takes the current count value, increments it, and returns the user to a refreshed copy of the HTML page with the new count displayed in it. The servlet resets the count value to 0 if the **Reset** button was clicked, and again displays a new copy of the page as above.

Problem 2: Storing multiple data values in a HttpSession

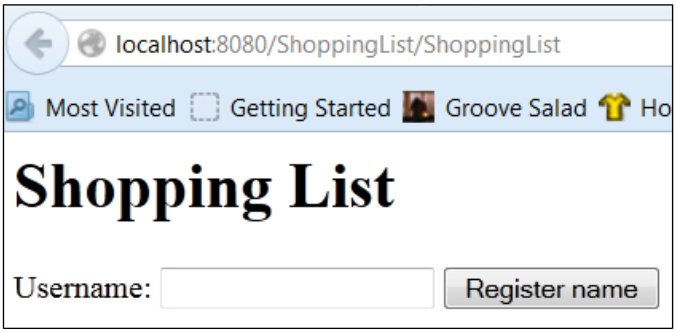
You are asked to create a web application called ***ShoppingList*** which allows a user to create a shopping list of items which will be displayed in the browser. The user will be able to register his/her name, and then add or delete items from the list.

The application must store the username and the list of items as an attribute of a ***HttpSession*** in the application.

The application should consist only of two servlets:

- **ShoppingList**: displays the username registration form and the shopping list form, as well as a Logout hyperlink. This should be the default object in the application
- **Validate**: Creates or destroys a ***HttpSession*** for the user as s/he logs in and out of the application

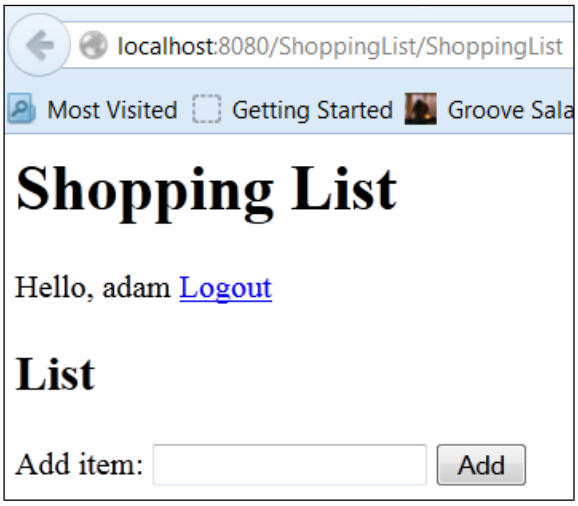
Example: user accesses application, sees ShoppingList page



Initially the application shows only a heading and a form allowing the user to enter his/her username (anything is allowed for the username). When the user clicks on the "***Register name***" button the name is submitted to a servlet which will create a **HttpSession** for the user and store the username provided in an attribute of it called "**username**". The user will then redisplay the ***Shopping List*** page.

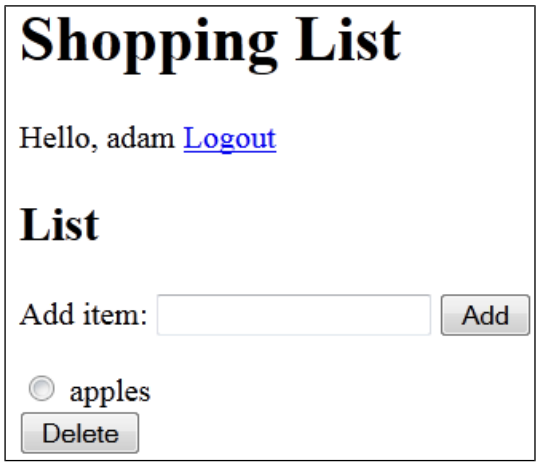
Note: we will consider registering a name to be "logging in" for this application.

Example: the user has registered a username



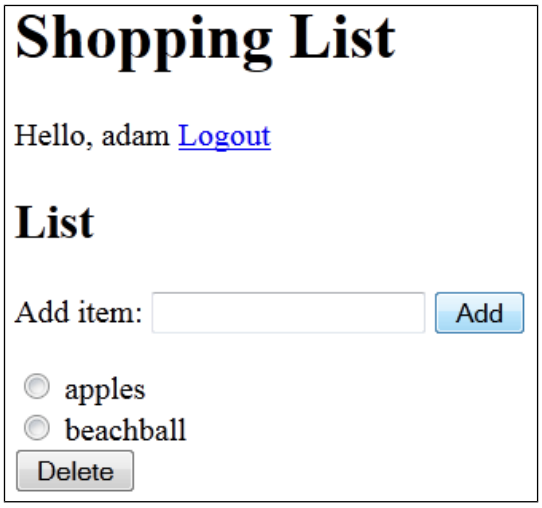
Because the user has registered a username ("**adam**") the ***Shopping List*** page now shows a welcome message (including the username), a "***Logout***" hyperlink, and a form allowing the user to add items to his/her shopping list.

Example: user adds "apples" to their list



Under the shopping list form "**apples**" is displayed. Beside it is a radio button that the user can select and then click on the "***Delete***" button to delete that item from the list.

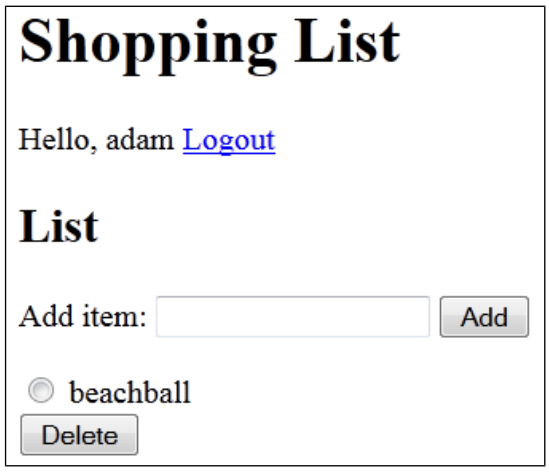
Example: user adds a “beachball” to the list



“**beachball**” is added to the bottom of the list. Further items will be added in the same way.

Example: the user deletes “apples” from the list

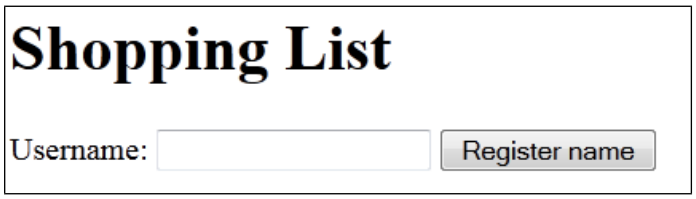
The user selects the radio button beside “**apples**” and clicks on the “***Delete***” button:



The user should be able to add and delete any items to/from the list as required.

Example: the user logs out of the application

The user clicks on the "***Logout***" hyperlink in the ***Shopping List*** page to leave the application. This link must send the browser to a servlet, sending it an additional parameter of "**?logout=true**". When the servlet runs it should check for the presence of a parameter called "**logout**" and if it exists it should destroy the **HttpSession** for the user and redisplay the **Shopping List** page.



Destroying the **HttpSession** resets the application.

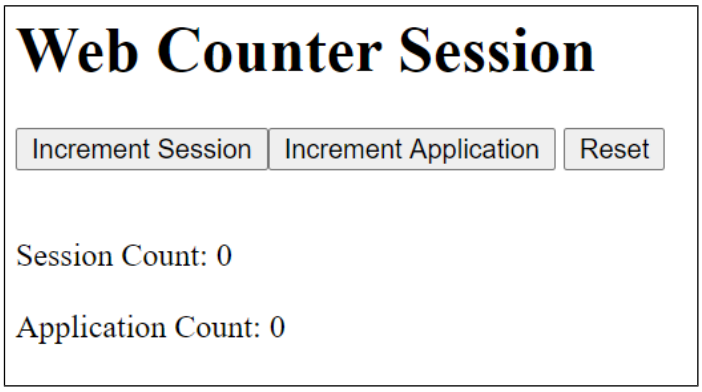
Application Requirement:

Both the username and list of items for the shopping list must be stored in the **HttpSession**. You can store an **ArrayList<String>** (or equivalent data structure of your choice) in the session object and use it to hold the shopping list items. A simple String can hold the username.

Problem 3: Revise solution for problem 1 to include application-level counter also

Make a copy of the **WebCounterSession** application you created for problem 1. Open the project in NetBeans and rename it to **MultiWebCounterSession** (remember to rename the directory the project is in to the same name also). In this new version of the application you will add another counter, this time one with application-level (servlet context) scope. Users will be able to increment and reset either the session- or application-level scope values as required. Use multiple browsers to test the application and ensure that the behavior is as expected, i.e. users share the same application-level count, but separate session-level counts. NOTE: using multiple tabs within the same browser may not work to simulate multiple simultaneous users, so you should use two or more different types of browers, e.g. Chrome, Edge, Firefox, etc., to test the application.

Add another button to the page in the application to let the user increment the additional application-level count value, and show the application-level count underneath the current session-level count value, e.g.:



As users click on the respective buttons they should see the count values increment appropriately. Clicking on the Reset button should reset both counts to zero.