Server-Side Web Programming

Assignment 3: Session Handling

Due Tuesday, March 10

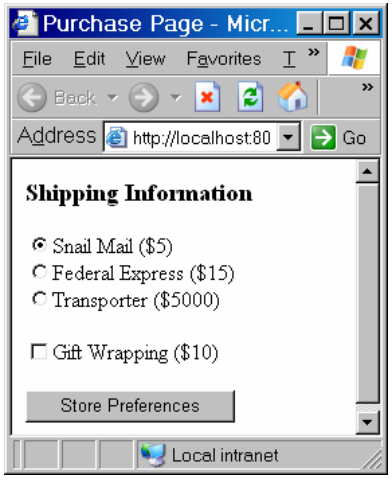
Introduction

You are to modify the "purchase site" you created in the previous assignments into a set of JSPs and servlets, using **session handling** to keep track of the items selected by the user, as well as the shipping and gift wrapping choices.

Ordering Process and Requirements

This process will have three stages:

- The user starts at **`index.jsp`**, where they enter their **shipping** and **gift wrap** choices.

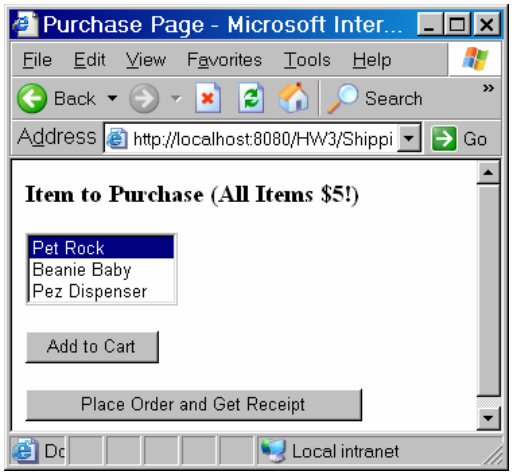


Yes, I know that this is usually done **last**. However, this will give you chance to get practice implementing something simple with sessions before you start working with a shopping cart, which is harder.

I have provided an **`index.jsp`** file for you to work with.

- When they press the **Save Preferences** button, this should take them to a **servlet** called **`ShippingServlet`** (which you will write). This servlet will store the shipping and gift wrap parameters in the **session** for the user.

- The user should then be forwarded to the **`AddItem.jsp`** page. This page lists the items they can choose from:



Note that *quantity* will no longer be entered. To simplify things, customers will no longer be allowed to order more than one of any particular item.

Note that there is a button which takes them directly to their bill (which is not usually done). However, you can use this to go to the bill directly without doing anything with the shopping cart, in order to test your initial work with the shipping/gift information.

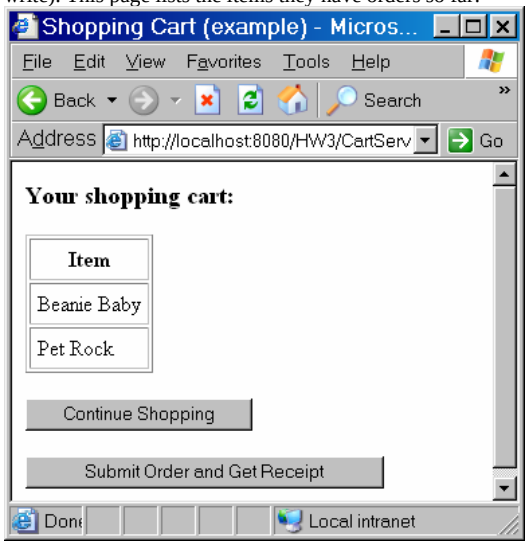
I have provided an **`AddItem.jsp`** file for you to work with.

- When the user selects an item from the list and presses the **Add to Cart** button, this should take them to a **servlet** called **`CartServlet`** (which you will write). This servlet will add the item selected to the **shopping cart** which is stored in the user's session. Your servlet must construct a **`Cart`** object and store it in the session if it does not exist.

I have provided **`Cart`** and **`Item`** support classes, which I will describe in more detail below.

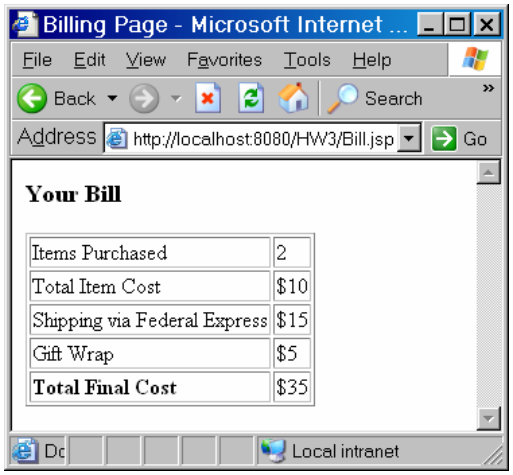
Note that I am not requiring you to do any validation for either the shipping method or the item selection in this assignment. The only validation that you are required to do is to **only add an item to the cart if it is not already in the cart**. You are not required to display any error messages in this case – just make sure that it is not added to the cart.

- The user should then be forwarded to the **`ShowCart.jsp`** page (which you will write). This page lists the items they have orders so far:



If they press the **Continue Shopping** button, they should be taken back to the **`AddItem.jsp`** page.

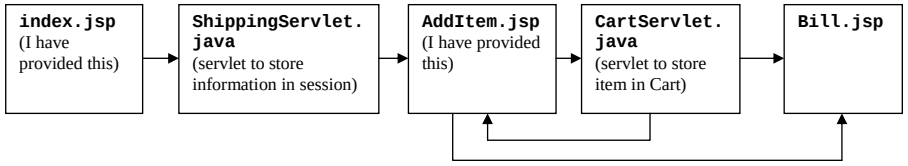
- If they press the **Submit Order and Get Receipt** button, they should be taken to **`Bill.jsp`**. This will be a modification of you previous bill pages, which displays
  - The number of items purchased
  - The total cost (at $5 per item)
  - The shipping cost
  - The gift wrap cost (if applicable)
  - The total bill



In order to do this, your JSP will need to retrieve the following from the session:

- The shipping method stored in the session by **`ShippingServlet`**
- The gift wrapping
- The number of items in the cart (I have provided a method for this)

In terms of overall site structure, the flow will look like this:



.

Session Support Objects

In order to keep track of the items the user has added, you will need to store them in a **session variable**.

As is usually the case in such systems, I have provided two "shopping cart" classes called **`Cart`** and **`Item`** to simplify this process. You will need to put them in a package called **Model** and import that package to the servlets and JSPs that use the cart.

The **`Cart`** class has the following methods:

- **`void addItem(String name)`**
  This method adds a new item with this name to the cart.

- **`Item lookup(String name)`**
  This method returns the item in the list with this name, or **null** if that item is not in the cart.

- **`int getSize()`**
  This method returns the number of items in the list. It is useful for creating a loop to display all items.

- **`Item getItem(int i)`**
  This method returns the i^{th} item in the list. It is also useful in a loop that displays all items.

- **`void removeItem(String name)`**
  This method removes the item with this name from the list.

Since we are only storing the names of items, the **`Item`** class only has the following method:

- **`String getName()`**
  This method returns the name of this item. It is useful in a loop wich diplays the name of all items.

Additional Requirements

Meeting the above requirements will give a score of **90%** (that is, 45/50) on this assignment. For full credit, you must do at least one of the following:

Redirection
It is possible that a user might access pages out of order (for example, they go to the **`ShowCart.jsp`** page without first having added an item at the **`AddItem.jsp`** page). If they access any pages out of order, you should redirect them to the initial **`index.jsp`** page.

Cookies
Your web site should still be able to keep track of a user's session even if they **don't enable cookies**. Use URL encoding to do this, and make sure that it works on all of your pages.

Removal of Items
Add a REMOVE button for each item in the shopping cart. Pressing it will remove the item. You will need a servlet to handle this, and will need to use an `INPUT` element of type **`HIDDEN`** to pass the name of the item to remove.

Shipping Support Object
A better idea than just storing the shipping/gift wrap data directly into the session would be to create a **business logic class** which:
- Stored the shipping and gift wrap method.
- Automatically computed the shipping and gift wrap amounts.

Implement this as a support class (adding it to your **Model** package) and use it instead of computing this information in **`Bill.jsp`**.

What to Turn In
You submit to **me** your **server page** and **servlet** files.

As with any other program you write, your server page must be **well documented**.