

Project Part 7 - Session Management

20/20 Points

7/20/2021

Attempt 1

**REVIEW FEEDBACK**

7/20/2021

Attempt 1 Score:

20/20

View Feedback

Unlimited Attempts Allowed

▼ Details

Due: Tuesday, July 20**Points:** 20 points**Deliverables:** [Name]BookstoreSession.war uploaded to CS5244 server and a submission link in Canvas

Project 7: Session Management

We are using a central Vuex store and can add items to our cart. In this part of the assignment, you will add a cart page and cart updates to your project. You will use the shopping cart object given to you in Project 7 to implement updating the quantities on your cart page.

Setup your project

Create a new (no-plugin) Gradle project in IntelliJ called [Name]BookstoreSession. Create two modules `server` and `client`, and copy files from your State project (P6) into the server and client sides. Remember to modify `vue.config.js` on your client side. Setup a Tomcat run configuration in your server project.

Try Again

Run the client using "npm run serve". The Home and Category pages, with the add-to-cart buttons and cart count in the header should look and behave just as they did in Project 6.

Create a cart page

Create at least two Vue components for your cart page:

- Cart.vue — under the views folder
- CartTable.vue — under the components folder

Create a (mostly blank) checkout page

Create a single Vue component for your checkout page:

- Checkout.vue — under the views folder

Please have a short message on the page so that there is some space between the header and footer. Feel free to have a button that takes you back to the cart page, but it is not a requirement.

Fill out the cart page

The cart page must have the following:

- A continue-shopping button that takes you back to the category page. Make sure the selected category on that


Try Again

- A proceed-to-checkout button that takes you to the empty checkout page
- A "clear the cart" button somewhere on your cart page that empties the cart with a single click.
- Some text stating how many items are in your cart
- Some text stating the subtotal for the items in your cart
- A table containing the items in your cart
 - Each row should have at least the book title, quantity, and unit price of the book. The total price (unit times quantity) is also recommended, but not strictly necessary. Any column that contains prices should be aligned on cents. Typically this means right-aligned.
 - Each row should have a way to update the quantity (for example, a field plus an update button, or decrement and increment buttons)
 - There should be a way to eliminate a book from the table (either through a single remove button or by taking the quantity down to 0)
- If there are no items in your cart, you should have a message saying that the cart is empty along with the continue-shopping button. You should not display that proceed to checkout button if the cart is empty, and you should not display an empty table if the cart is empty.

You may implement the table any way you like, although we recommend using HTML table elements for simplicity. Also, you are responsible for knowing how to use HTML tables. CSS-based tables are similar to HTML tables. The web designer of "A Bookstore" uses CSS-based tables.


Here are some sample cart pages that you can use for inspiration. Notice that some of these pages do not strictly meet all the style requirements given below, so make sure you take a look at those and ask on Piazza if you have any questions about them.

Try Again


 4 items

[proceed to checkout →](#)

english [česky](#)






the affable bean



Your shopping cart contains 4 items.


[clear cart](#) [continue shopping](#) [proceed to checkout →](#)


subtotal: € 10.68


product	name	price	quantity
	cheese	€ 2.39 (€ 2.39 / unit)	<input type="text" value="1"/> update
	sesame seed bagel	€ 1.19 (€ 1.19 / unit)	<input type="text" value="1"/> update
	sausages	€ 7.10 (€ 3.55 / unit)	<input type="text" value="2"/> update

[Privacy Policy](#) :: [Contact](#) © 2010 the affable bean

[Try Again](#)

 Greater Goods Grocers





Welcome, Guest
[Sign In](#) | [Register](#)

Fresh ProduceMeatDairyBakery

Your shopping cart contains 2 items.




Product Name	Description	Points	Price	Quantity	Total
Croissants		2	\$2.49	1	\$2.49
Doughnuts	assorted flavors	3	\$1.29	1	\$1.29

Cart subtotal: \$3.78
Cart total: **\$8.78**

Clear CartContinue ShoppingProceed to Checkout

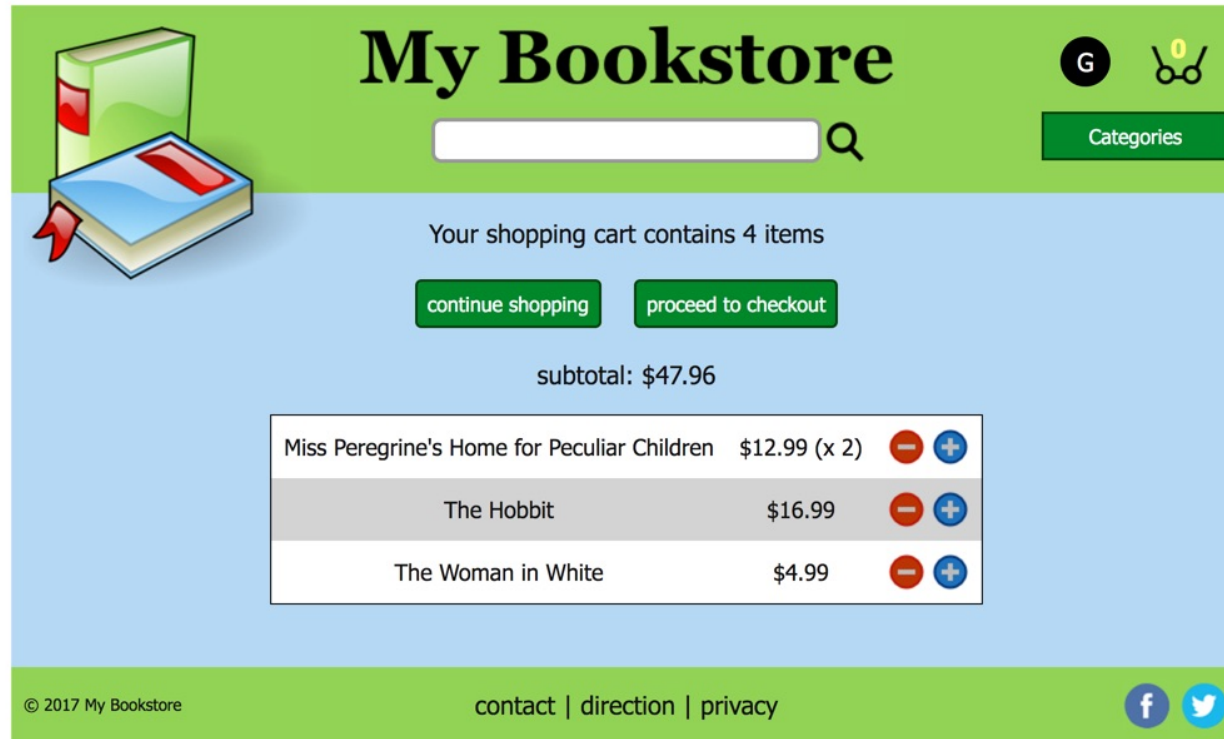
Store Hours: 7 days a week. 6am-10pm PST

[Contact Us](#) | [Privacy Policy](#) | [Terms of Use](#) | [Directions](#)

Copyright (c) 2019 Greater Goods, Inc.

Try Again



My Bookstore

Your shopping cart contains 4 items

[continue shopping](#) [proceed to checkout](#)


subtotal: \$47.96


Miss Peregrine's Home for Peculiar Children	\$12.99 (x 2)	-	+
The Hobbit	\$16.99	-	+
The Woman in White	\$4.99	-	+

© 2017 My Bookstore [contact](#) | [direction](#) | [privacy](#) [f](#) [t](#)

[Try Again](#)

A Bookstore

A Bookstore
Find a book... 


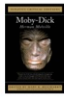


 (0)

Sign In

Categories




Your Cart
Items: 5

Continue Shopping Checkout

	The Brothers Karamazov (or another title with a particularly long name)	1	\$12.99
	Moby Dick	1	\$14.99
	Little Dorrit	1	\$9.99
	Pride and Prejudice	2	\$13.99
Sub-total:			\$65.95

Continue Shopping Checkout

Contact Us Directions Privacy

© 2017, A Bookstore

Save your cart updates to local storage

Create and export a string constant called `CART_STORAGE_KEY` in `store.js` and set it to "cart".

Create a mutation called `SET_CART` in `store.js`.

Try Again

```
SET_CART(state, shoppingCart) {  
  localStorage.setItem(CART_STORAGE_KEY, JSON.stringify(shoppingCart));  
  let newCart = new ShoppingCart();  
  shoppingCart.items.forEach(item => {  
    newCart.addItem(item.book, item.quantity);  
  });  
  state.cart = newCart;  
}
```

Add the following line to every mutator that modifies the cart:

```
localStorage.setItem(CART_STORAGE_KEY, JSON.stringify(this.state.cart));
```

Let us ensure that if the browser has a stored cart, we load it when a client page is loaded. In `main.js`, import the cart storage key from `store.js` and add the following to your Vue instance:

```
created() {  
  const cartString = localStorage.getItem(CART_STORAGE_KEY);  
  if (cartString) {  
    const shoppingCart = JSON.parse(cartString);
```

Try Again


```
}  
}
```

Consolidate your currency code

Using a Vue filter (<https://vuejs.org/v2/guide/filters.html>) or a Price component (for example in GreaterGoodsGrocers) format your book prices in CategoryBookListItem and on the cart page.

In each case make sure you use the international number formatter (see <https://flaviocopes.com/how-to-format-number-as-currency-javascript/>).

```
// From https://flaviocopes.com/how-to-format-number-as-currency-javascript/  
const PriceFormatter = new Intl.NumberFormat('en-US', {  
  style: 'currency',  
  currency: 'USD',  
  minimumFractionDigits: 2  
})  
  
Vue.filter('asDollarsAndCents', function (cents) {  
  return PriceFormatter.format(cents / 100.0)  
})
```

Try Again

Style Requirements

In addition to the functional requirements given above, here is a list of mostly stylistic requirements that should be observed. The list includes a repetition of some functional requirements based on common problems observed in past courses.

Empty cart

- Don't put an empty table on the cart page
- Don't put \$0.00 for the price, and don't put "subtotal" if the cart is empty
- Don't put "0 books", say "your cart is empty" or something similar
- Don't say "1 books", say "1 book"

Cart table

- Make sure all table cells have padding
- Don't put the quantity between increment or decrement buttons unless the buttons are smaller than the font used for the quantity
- Always align prices on cents (right-align)
- If table has a book image, ensure that the table's book image is smaller than the image on the category page

Cart icon

- Make sure all fonts large enough to read, including the font used for the number of items in the cart
- Make sure both the cart icon AND the cart-count link to the cart page. The cursor should turn into a pointer when you hover over the cart icon and/or the cart-count

Cart buttons

- Don't put the clear cart button in same row as continue-shopping and proceed-to-checkout
- If continue-shopping and proceed-to-checkout ARE in same row, make sure continue-shopping comes first
- Make the "proceed to checkout" button more prominent (in style and/or color) than the continue shopping button
- Make sure the continue-shopping button takes you back to the last category you were on
- Align your buttons appropriately with respect to the table

Buttons (general)

Try Again

- Make sure buttons or links subtly change style when you hover over them
- In general, buttons should not change size when you hover over them (for example, if you are changing the font weight to bold)
- If buttons change size because you want a certain effect (like sliding out), the effect should not change the position of other elements on the page

Professional web designer's bookstore

You've already seen the home page and category page that our professional web designer (Chris Blessing) came up with. This zip archive also includes the cart page, checkout page, and confirmation page. As before, if you decide to use any code, make sure you modify it to make it fit with your site.

- **[A Bookstore ZIP file \(https://canvas.vt.edu/courses/132380/files/18823416/download?wrap=1\)](https://canvas.vt.edu/courses/132380/files/18823416/download?wrap=1)**

<http://cs5244.cs.vt.edu:8080/GraceBookstoreSession/>

Try Again