COM S/SE 319: Construction of User Interfaces
Fall 2022

Homework 2: Store Cart

[Total Points: 50]

Assignment Due: Friday, Sep 30th, 2022, 11:59PM

1. Overview

In this assignment you will develop a small cart application. The cart you'll build is somewhat of an extension of the lab 04 exercise. Your client is a small store that sells a variety of products. The client requires 3 views.

1. A browse view
   a. Contains a finite list of "addable" items.
   b. User can pick multiple types of items and 1 or more instances of each item.
2. A cart view
   a. Displays the list, quantity and value of selected items.
   b. Displays the total value + tax.
   c. A checkout form to accept user credit card information and shipping address.
3. A confirmation view
   a. A page that shows the order summary.

We want all three pages to be implemented using the single-page design. I.e. you will only need one index.html page and ideally one other script.js page. You can also add your own style.css page. But custom styling is optional. We expect you to use bootstrap.min.css and bootstrap.min.js files. You can also use the bootsrap icons css file. To better understand the single-page design, checkout Lab 04 and see how to transition from one view to the next. Specifically, how we implemented event listeners and toggled the ".collapse" class.

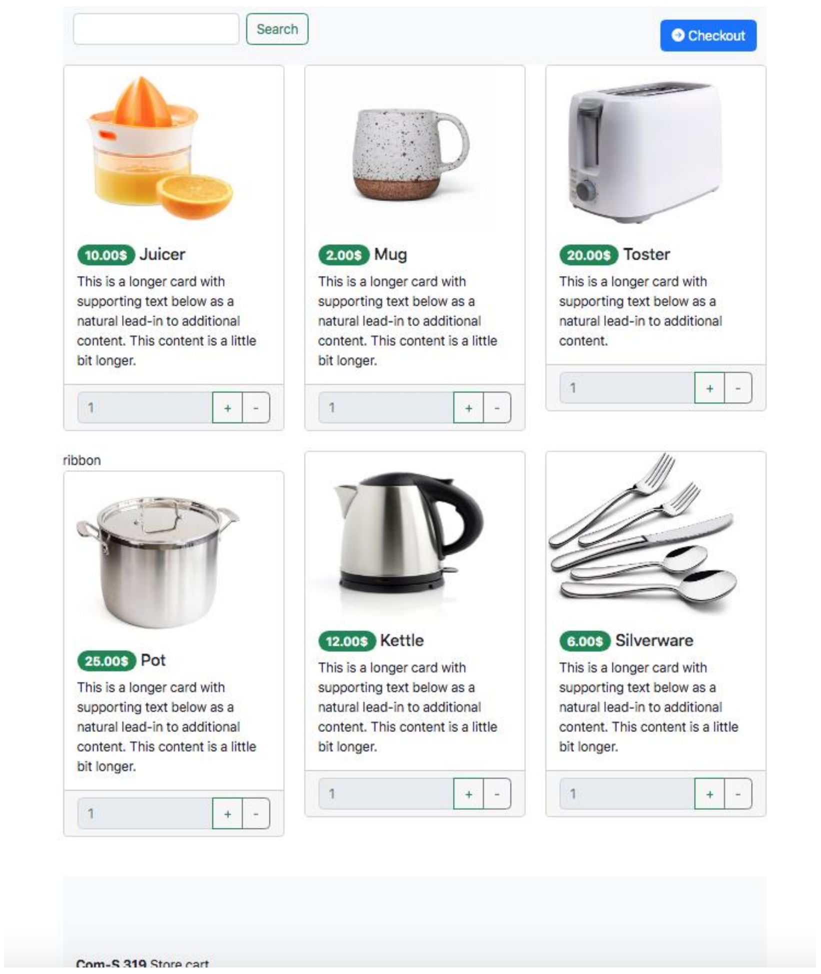
Feel free to add small touches to the webpage to enhance the view. You can use:

- Bootstrap icons, components etc.
- Feel free to play with font sizes, colors. Esp with numeric values.
- You can customize the browse view, table and form components however you like. But make sure you display all stated fields and values.
- Custom CSS to:
  - Enhance the look and feel with custom coloring, components, fonts etc.
- Just remember to maintain the underlying bootstrap theme. Similarly, make sure your customization doesn't get in the way of the expected functionality outlined below.

2. Design

2.1 Browse view (25 points)

- Your store must have a finite set of items. Anywhere between 6 –12 will suffice. In this example, the kitchenware store only sells 6 kinds of items.
- Feel free to sell other products.
- Notice how each item has "+" and "-" buttons directly below it. Whenever you click the buttons, the quantity in the adjacent input field should increment and decrement accordingly. Assume there is no upper limit to your item count, but it shouldn't be a negative number. So watch out not to decrement below 0.
- Make sure you update a global "cart" variable every time item quantities change.



10.00$ Juicer
This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

2.00$ Mug
This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

20.00$ Toster
This is a longer card with supporting text below as a natural lead-in to additional content.

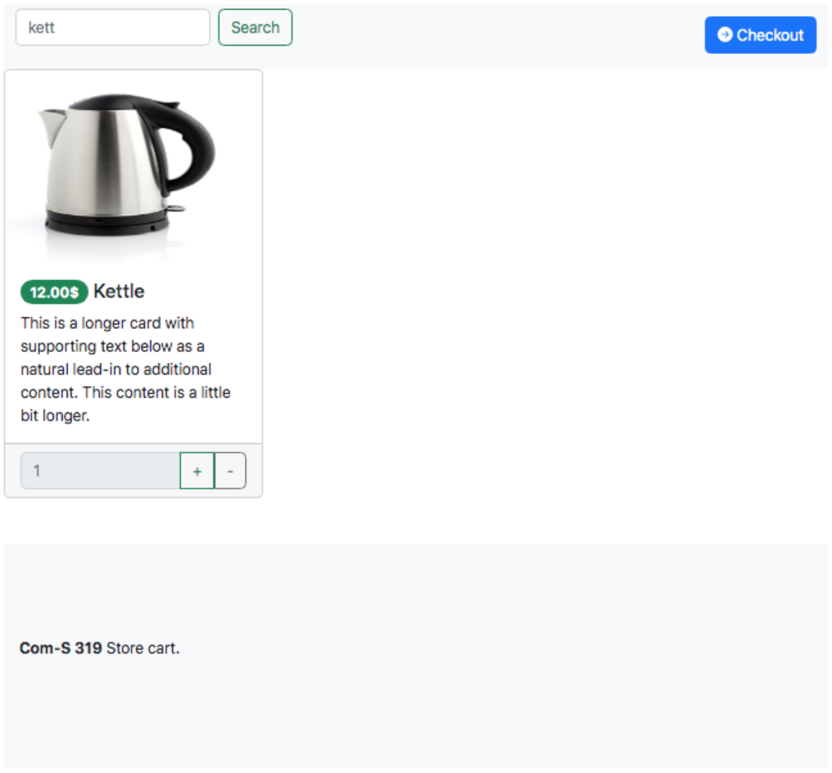
25.00$ Pot
This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

12.00$ Kettle
This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

6.00$ Silverware
This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

Com-S 319 Store cart

- Notice the search bar at the top. It should be functional.
  - Whenever a search pattern is inputted, all search cards containing the search substring as title should appear. At the same time, those whose titles don't contain the matching substring should fade away.
  - Feel free to add an additional "clear search" button.
  - In the example use case shown below, notice how typing the search substring "kett" is enough to filter the "Kettle". Note that search strings are case **insensitive**.
  - **[Hint: Listen to the "input" event on the search input field.]**

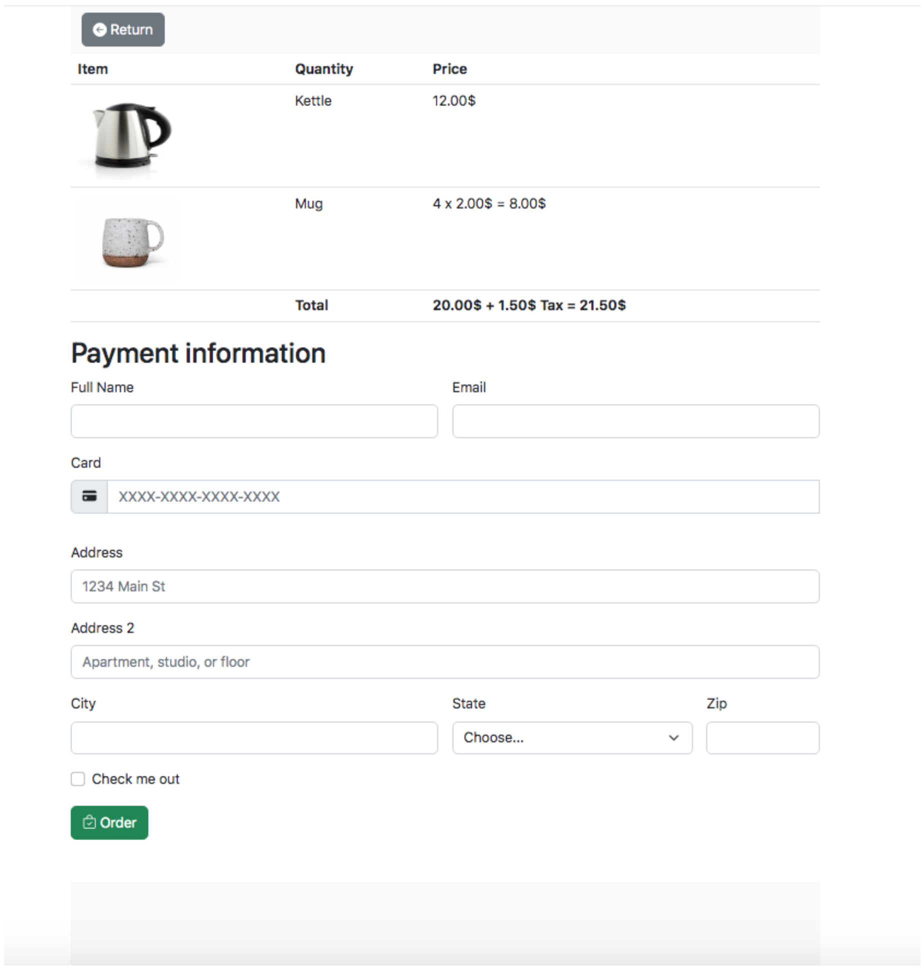


This is a longer card with supporting text below as a natural lead-in to additional content. This content is a little bit longer.

Com-S 319 Store cart.

2.2 Cart view (15 points)

- When clicking the “checkout” button, the user will be transported to the cart view. Remember we’re doing single-page design, so there is no page navigation, just a <div> swap.



Return

Item	Quantity	Price
	Kettle	12.00$
	Mug	4 x 2.00$ = 8.00$
	Total	20.00$ + 1.50$ Tax = 21.50$

Payment information

Full Name

Email

Card

XXXX-XXXX-XXXX-XXXX

Address

1234 Main St

Address 2

Apartment, studio, or floor

City

State

Choose...

Zip

Check me out

Order

- The cart view has a return button. When clicked, it should return us to the browse view.
    - Note that upon return, we should see all the items listed (no previous filters) with the correct quantity values in place.
- The cart view must show a list of selected items alongside their quantities and prices.
    - Don’t forget to add up the total value and tax.
    - Choose whatever tax formula you want.
- Finally the page must have a payment form.
    - Full name, email, card, address1, city, state, zip are all required fields! Furthermore they require validation!
        - Validate email and credit card fields. Also validate zip code. i.e. make sure it only accepts 5 digit numeric strings. The other required fields need to have an arbitrary string value to be considered valid.
    - Address2 is an optional field.

    - The checkbox serves no purpose, so feel free to remove it.
- When clicking “order”, we should navigate to the confirmation view.
- Remember to test your navigation robustness. I.e:
    - If you click “return”, you’ll go back to the previous browse view with the correct quantities set and no search string in the search field!
    - Click “return” followed by “checkout” and you should land back on the same cart view. (i.e no changes in quantities or prices)

2.2 Confirmation view (10 points)

- We leave the design of the confirmation view up to you.
- The only requirement is to display the following info:
    - Purchased items, preferably with pictures, and total purchase amount, optionally with price breakdown.
    - User information (Preferably redacting the first 12 digits from the user’s credit card)
    - A button that will navigate you back to a fresh browse view. Don’t forget to reset your “cart” object!
- Tips on designing the confirmation page:
    - Stick with the underlying bootstrap theme.
    - Season the view with whatever custom style you think will enhance the look!

3. **What to Submit:**

- Include these files
    - index.html
    - script.js
    - Images
    - [optionally your style.css if you have one]
- Compress the folder to a (.zip) and rename it to your “LastName_FirstName.zip”.
- Submit via Canvas the **compressed file (.zip)**.