IS216 AY2025/26 Term 1 Trial Lab Test Set B
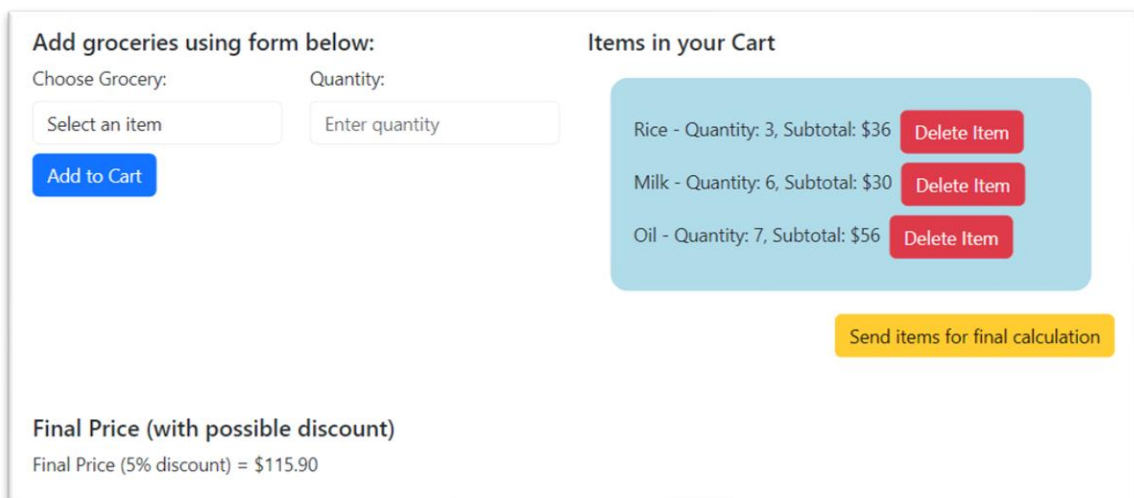
Bootstrap, JavaScript and AXIOS

Given resources in folder `Q3`:

🚫 `q3.php`

✏️ `q3.html`

✏️ `q3.js`

Instructions

1. Your solution will be graded based on the correct output and behavior of your HTML page. No partial marks will be given for incomplete/incorrect solution.

2. The following screenshot is an example of what the **final working app** should look like at >= 992 px.

3. Part A is independent of Parts B, C and D. i.e. even if the Bootstrap question part A is not done, you can still do Parts B, C and D, and vice versa.

Part A:

1. When the browser width is **>= 992 px (lg breakpoint)**, the form should appear as shown below:

- "`Choose Grocery`" and "`Quantity`" fields should occupy *3 columns* of Bootstrap 12-columns grid, each.
- "`Items in your Cart`" div should occupy *6-columns* of Bootstrap 12-columns grid.
- The "`Send items for final calculation`" button should be justified to the right.

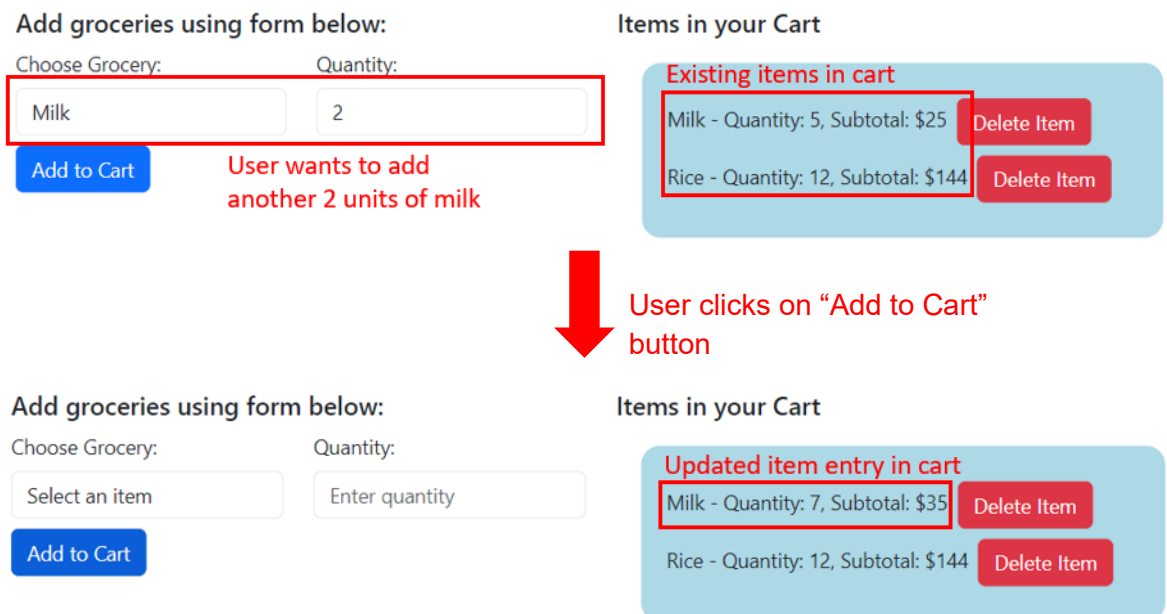2. When the browser width is **< 992 px (lg breakpoint)**, the form should appear as shown below:

- "`Choose Grocery`", "`Quantity`" and "`Items in your Cart`" fields should occupy the full width of the screen.
- The "`Send items for final calculation`" button should be centralized.

Note:
- You are only required to **add Bootstrap classes** in `q3.html` to achieve the desired page layout.
- You are **NOT** to use JavaScript to achieve the web page layout required.
- You are not required to add/edit other styles, such as margin, padding, font style, font size, etc., not mentioned in the requirements.

Part B: When the user selects an item from the dropdown list of "`Choose Grocery`", as well as input a non-zero quantity, a div should be added under the div of id="`cart`", with the following specifications:
- The added div should have an id of "`rice`", "`milk`" or "`oil`", depending on which item is added.
- The text to display in the added div should look like these examples: e.g.
    - Milk - Quantity: 5, Subtotal: $25
    - Rice - Quantity: 12, Subtotal: $144
- After adding an item e.g. 5 units of milk added, if the user adds the same item again, the corresponding div should be updated with the new quantity and subtotal, instead of having another div added i.e.

Note:
- You are only supposed to **add vanilla JavaScript** in the `addToCart()` function in `q3.js`.
- You are NOT to modify the HTML portion in anyway, nor are you required to.
- Use the items array defined at the top of the `q3.js` file to get the price of each item.

Part C: Add a "`Delete Item`" button in the div element created for every new item added to the cart. When the user clicks on the "`Delete Item`" button, it should delete the item's div element from the "`cart`" div.

Example hierarchy (with incomplete HTML, showing only hierarchical relationship):
- `<div id="cart">`
  - o `<div id="milk>`
    - ▪ Text e.g. Milk - Quantity: 7, Subtotal: $35
    - ▪ `<button>Delete Item</button>`

For styling, use the Bootstrap classes "`btn btn-danger ms-2`" for your "`Delete Item`" button.

Note:
- You are only supposed to **add vanilla JavaScript** in the `addToCart()` function in `q3.js`.
- You are NOT to modify the HTML portion in any way, nor are you required to.

Part D: Complete the "`showFinalDetails()`" function in `q3.js` to accomplish the following:

i) Send the items' quantity and subtotals in the cart to "`q3.php`" in a "`data`" object, through a **POST** request.

ii) Display the final total of the items in the cart in the div with id="`finalPrice`"
   a. with the text: "`Final Price = $...`" **if discountReached (to be retrieved from the response object returned from the POST request) is false**.
   b. with the text: "`Final Price (5% discount) = $...`" **if discountReached (to be retrieved from the response object returned from the POST request) is true.**

Examples of the final output are shown below:
- **If discountReached is false:**

- **If discountReached is true, apply a 5% discount to final price:**

Note:

a) This is how "q3.php" expects your POST request:
- A "data" object containing an array of objects should be sent, each with **at least one key-value pair of**

  { subtotal : Actual item subtotal (dollar sign not required) }

  e.g.
  ```
  [
      { subtotal : 35},   // for milk item of quantity = 7, subtotal = 35
      { subtotal : 32},   // for oil item of quantity = 4, subtotal = 32
  ]
  ```

b) This is how "q3.php" will send back a successful response:
- An object containing the following key-value pairs:
  - discountReached : true or false (based on whether the sum of subtotals >= a threshold of $100)
  - finalTotal : a number containing the sum of subtotals, with a discount of 5% applied if the sum of subtotals >= a threshold of $100

c) "showFinalDetails()" function will be called when the "Send items for final calculation" button is clicked on. This feature is already coded for you.

d) You are only supposed to **add vanilla JavaScript** in the showFinalDetails() function. You are NOT to modify the HTML or CSS portion in any way, nor are you required to.

e) A different PHP file with different discount threshold and percentage will be used to test your code, so you have to use q3.php to answer this question, instead of "hard-coding" your answer without using the PHP script's response.

END