JavaScript Final Project Assignment

Grading Note: Grading will be based on whether your page does what it is supposed to do, but also that it is written properly with correct HTML 5, CSS, and JavaScript. If any of your assignment parts isn’t working right but you are turning it in anyway, be sure to indicate that it doesn’t work properly. I will discover this anyway, but you might save me a lot of time and keep me in a better grading mood!

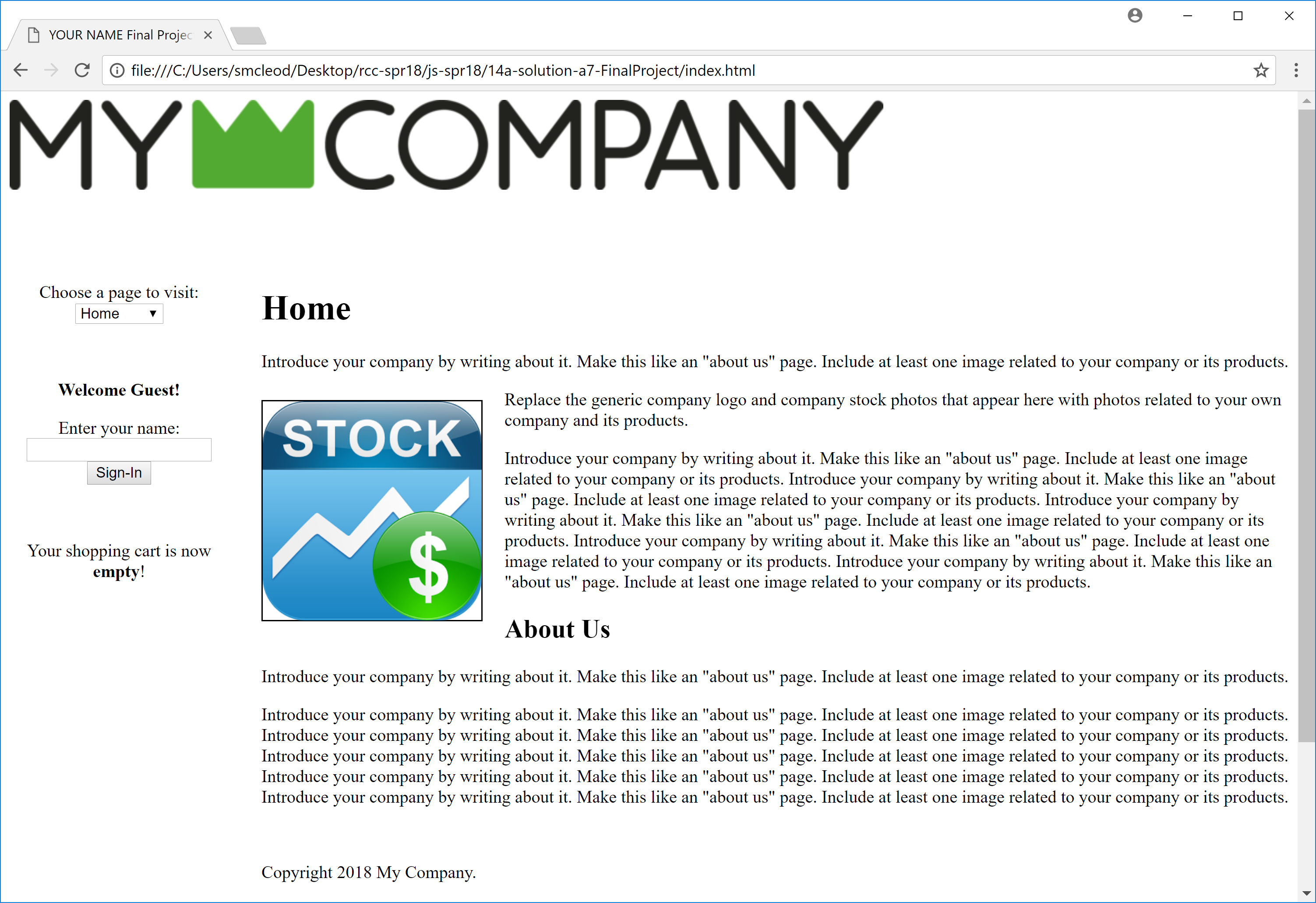
**Section 1: General Description**

Get your hands on the starter files for this project, which are given to you as a ".zip" download from within Canvas. Once you have this zip file on your local drive, unzip it and examine its contents. Amongst the many files, you should see 4 web pages: a home page, a shopping page, a cart page, and a checkout page. There are also various image files, a style sheet file, and javascript (.js) files. Familiarize yourself with this folder of files and their contents in File Explorer, a browser, and an editor. You should replace the given generic images throughout the site with your own that fit your own company and your specific products. Note that there are some helpful hints that were left for you within these given pages to help you fill in the gaps in the code.

**Section 2: Sample Screen Shots**

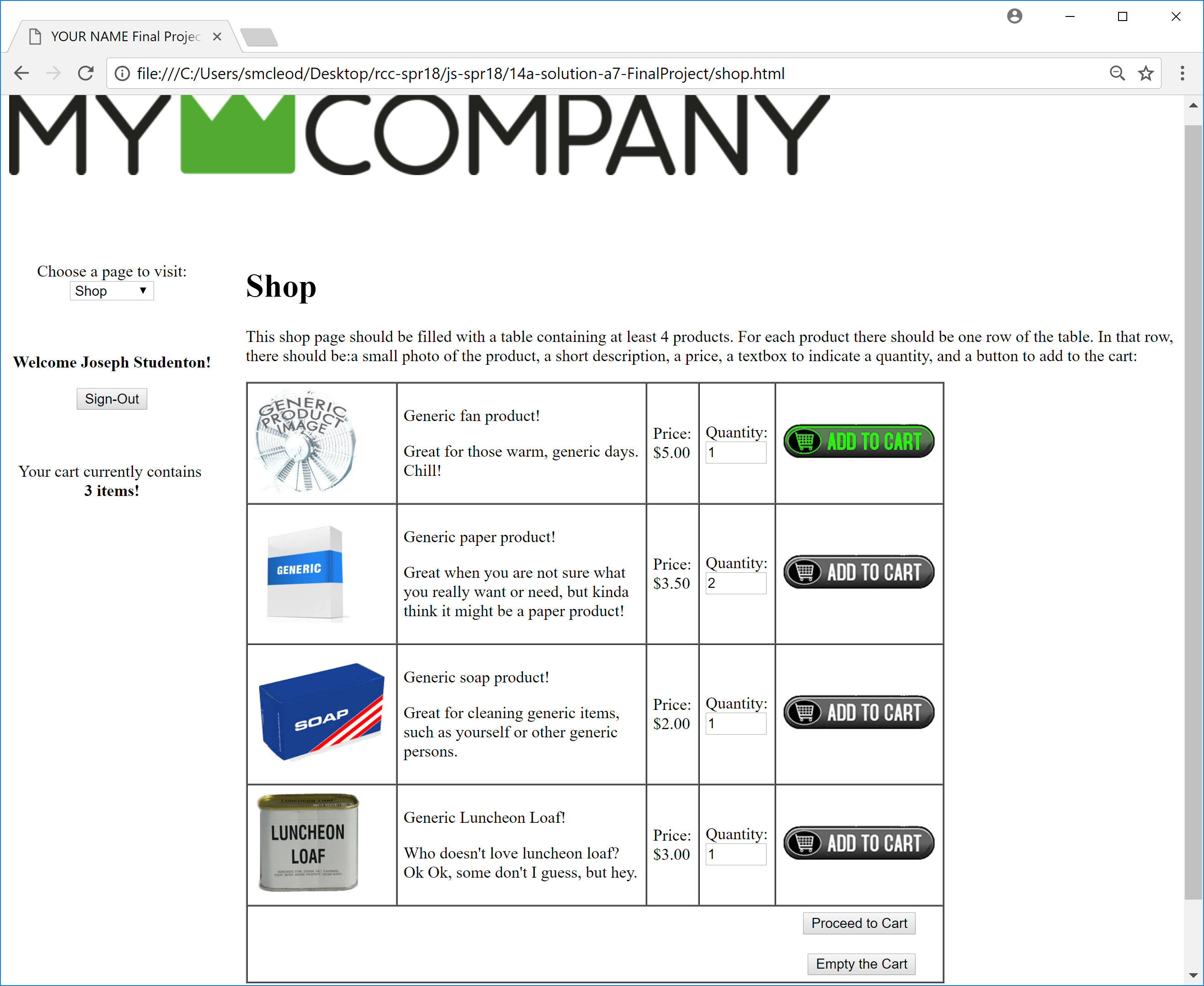
Here is what each of the 4 pages should approximate:

**1. Home Page**



**2. Shop Page**

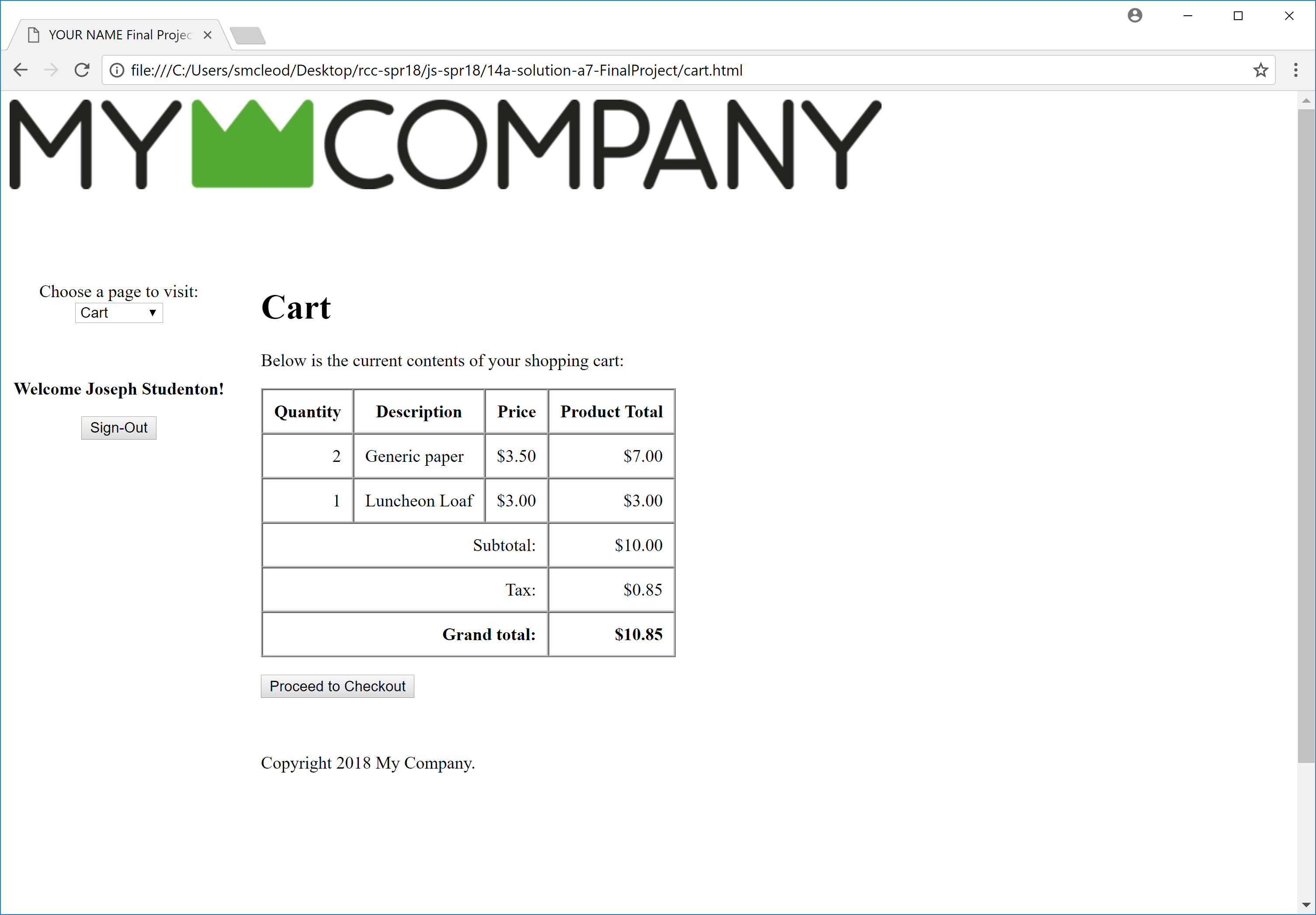
Create your own 4 products to list here on your products (shop) page. You can use the 2 you already have been using for earlier assignments if you like and then you just need to find 2 more. You can copy/paste some of that code from those previous projects to the degree it helps, but be careful to only copy over the parts of the pages that are appropriate. The cart page screen shot was taken after adding 2 Generic Paper products and 1 Generic Luncheon Loaf product (3 items total added to the cart) and then the mouse was placed over the Generic Fan Product's "Add To Cart" button (which is why the green colored button is showing). Note that you should write (HTML and/or JavaScript) code as needed to make sure that only appropriate quantities can be added to the cart (no negative numbers, no zeros, and no numbers greater than some reasonable max, like say 1000. For extra credit, you can produce this entire table using JavaScript with all of the table data (using filename strings for images) coming from your array of product information. Notice also the "Empty Cart" and "Proceed to Cart" buttons below the form. Those buttons should result in calls to functions that manage the cart in the given .js file.



**3. Cart Page:**

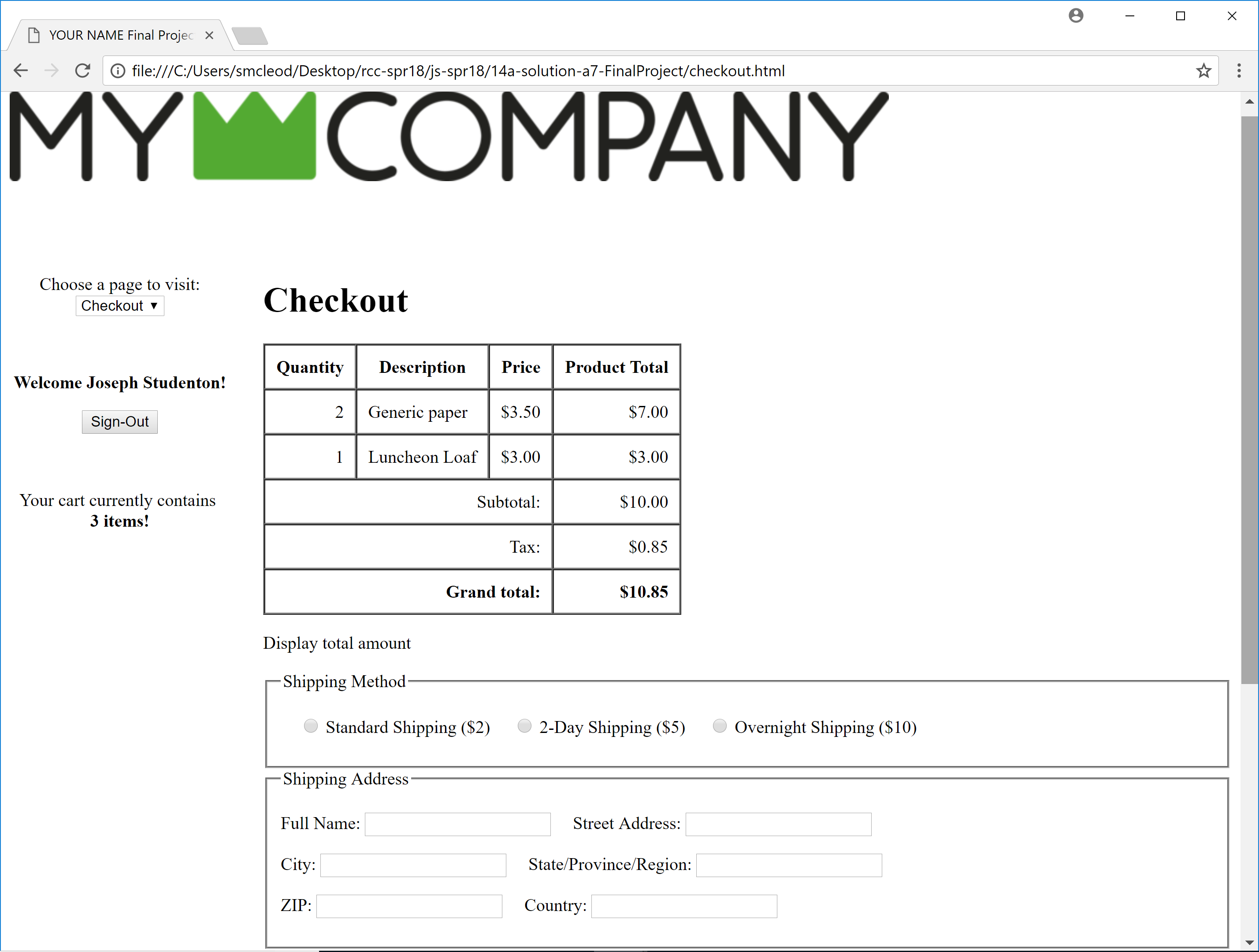
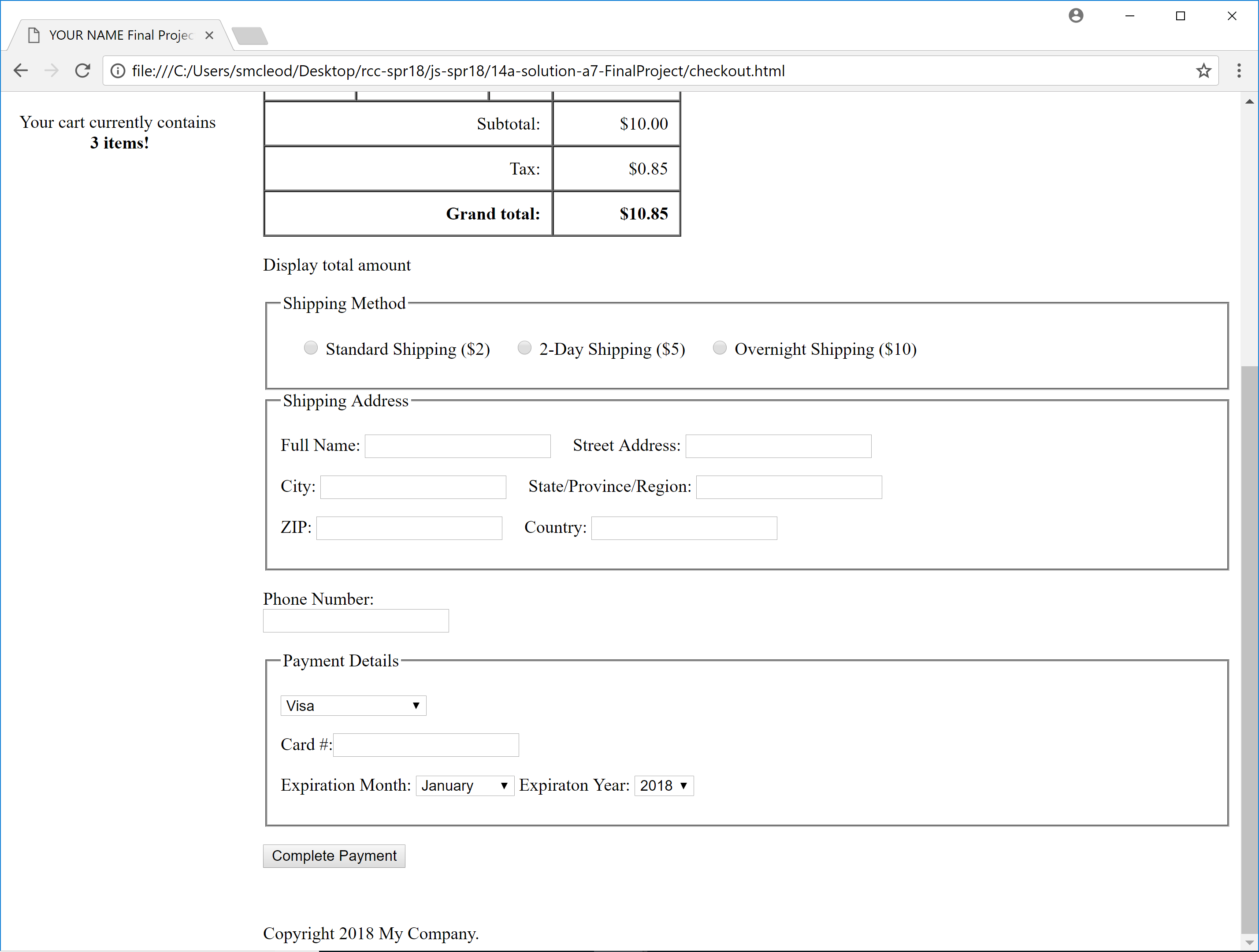
This page shows the current status of the shopping cart based on current quantities that you have in local storage for any products that have been added to the cart.

Extra credit can be given for additional functionality given to this page to allow easy editing of the quantities of each product in the cart. After any editing is done, local/session storage would have to be updated and the updated cart would need to be re-displayed.

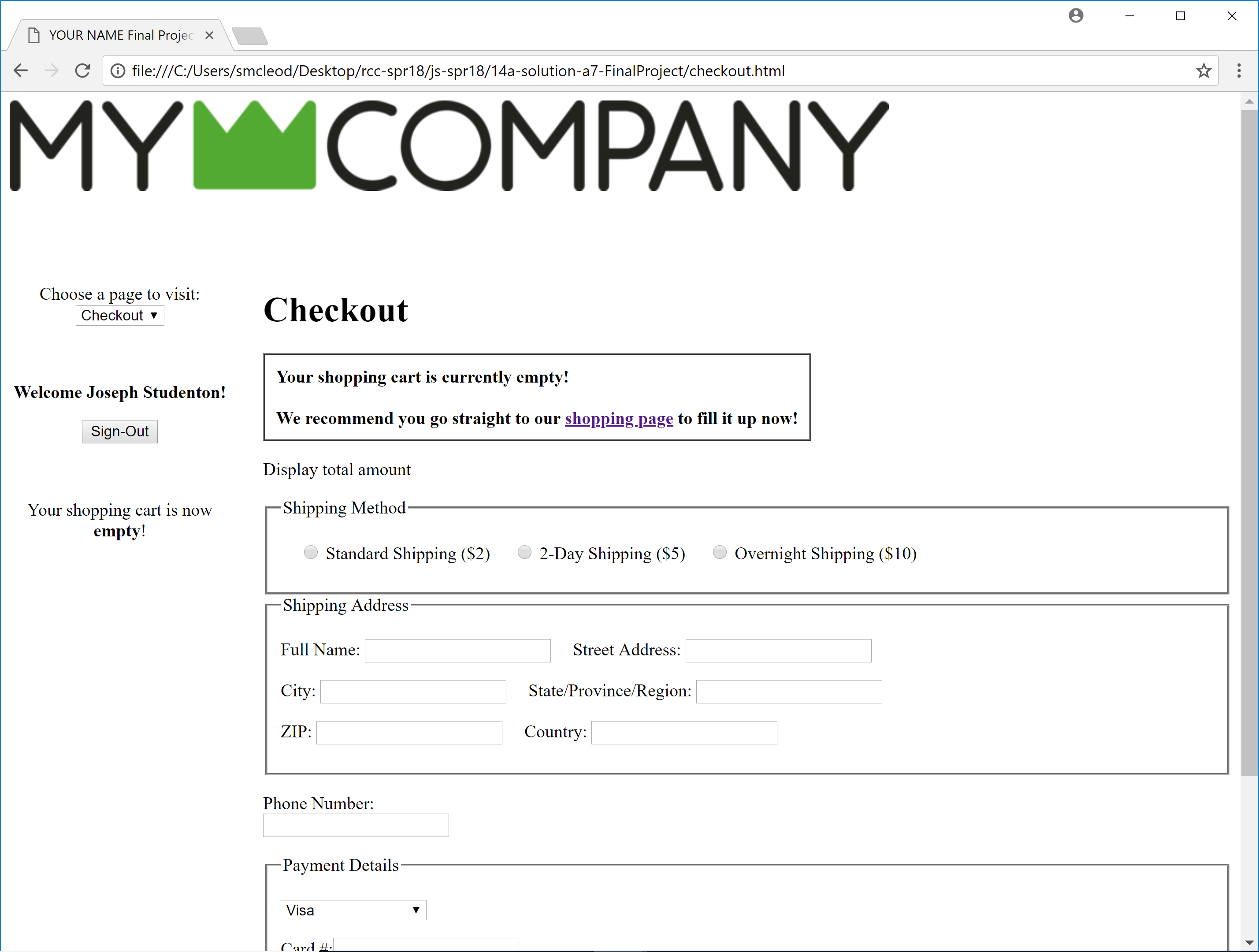


**4. Checkout Page**

This page would have some of the same form elements as your assignment pages have had such as billing and shipping choices, payment method, phone number, and final receipt. Below is an example of this form page. You will have to create the form yourself. Some validation of these form elements should be done so that the user must enter at least those elements that are critical to the transaction. Use HTML to help you validate, but some will have to be done using JavaScript too. Assuming the form has been completed properly, submitting it should take you to a "thank you" page. Note that if the cart is empty, there should be a statement of this fact and a suggestion to do some shopping instead of the empty cart display with $0 grand total. Use an 8.5% tax rate (constant) in your calculations.

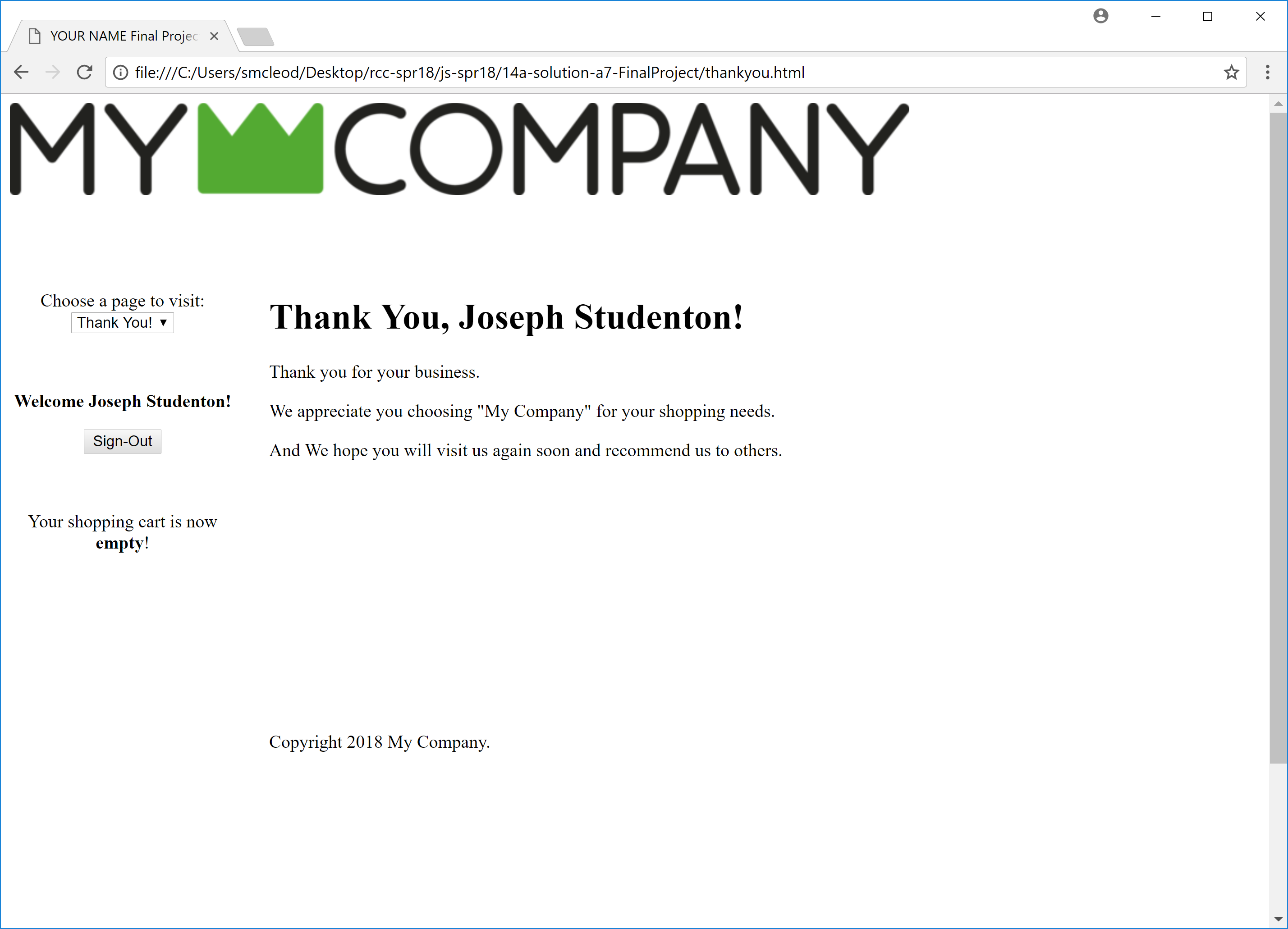
  


Here is an example of what the Checkout page might look like if the cart is empty. Notice the top area where there is a statement about the empty cart and a link to the shopping page.



**5. Thank You Page**

This page would be displayed after completing the payment form properly and submitting it. Below is an example of this page. Notice that the cart has been emptied. Even better if this page displayed more information from the transaction such as the items purchased and/or amount paid, almost like a receipt. Note that this page does not exist in your given pages, so you can copy a appropriate existing page and modify it as needed.



Work on these pages based on what is shown here (as a goal) and what is available in the downloaded files (as a starting point) and as further discussed in class. Also, there are comments within the .js files with some code that has been written for you and some code you need to fill in. If you have any questions, you can email your instructor or post in the Canvas discussion area.

**What to turn in:**

You will submit a zip archive of all of your files (including images, style sheets, javascript files, web pages, etc.) that make up your site, much like the zip file you download to begin your work. In Windows, this can be done in File Explorer by right-clicking the folder containing all of your site's files (this folder should have your name on it) and then choosing "Send to" and then "Compressed (Zipped) Folder". That is the file to submit into Canvas for your final project.

**Section 3: Grading Criteria:** 5 Basic Features: 200 Points + 5 Extra Credit Features: 50 Points

**1. (30 Points) Non-JavaScript content – HTML, CSS, Images**Should include a logo on each page, a company related image and text on home page, product images and descriptions on Shop page, etc. Includes some basic CSS to set fonts and page background and text colors.

**2. (30 Points) Navigation using the given Select List**Should function properly across all site pages, should display current page, should take customer to chosen page.

**3. (50 Points) Login/Logout Guest/Name Management**Should function properly across all site pages, should display "Guest" if no name is currently logged in, should update its display immediately upon a login or logout, should remember its status from page to page. When you get to the Checkout process, a logged in customer should have his/her name filled in for them in the "name" textbox. Code for these login functions should be in the given separate .js file.

**4. (50 Points) Cart Management**The shopping cart should maintain itself from page to page and should update itself when changes are made on the shopping page (either adding to the cart or clearing from the cart). Its status should appear in the navigation sidebar area, such as the current number of items (and maybe current dollar total) in the cart. "Add to cart" buttons on shop page should show "rollover" behavior when mouse hovers over them (should change buttons on mouse over and mouse out). After the purchase has been made during the checkout process, the cart should be emptied. Code for these cart functions should be in the given separate .js file (or in another one if you prefer to write your own.)

**5. (40 Points) Checkout Process**The checkout process should involve filling out billing & shipping choices and address details, as well as showing a final price calculation including tax and typical credit card input. Fields critical to the transaction should be validated either in HTML or JavaScript as needed to ensure they are filled out reasonably. Once the "transaction" has been made (form submitted successfully), take the user to a "thank you" page thanking them (by name) for their purchase.

**Extra Credit Features: 50 Points Possible**

**1. (10 Points) Non-JavaScript enhancements**Outstanding work on design with CSS, Logo, button images, HTML content, extra pages, etc.

**2. (10 Points) Use of Objects**Proper usage of your own JavaScript objects for the shopping cart or other enhancements. You can use some variation of the given cart code to design your own cart /cart items with properties and methods. It has to work!

**3. (10 Points) Cart Management Enhancements**The ability to make cart changes from the cart and/or checkout pages, such as to modify the quantity of any product in the cart, to delete all of any product currently in the cart, to empty the cart, etc. These should be one-click or minimum effort for the customer and should result in updating of all cart displays and purchase amounts.

**4. (10 Points) Checkout Process**Enhanced checkout process, including enhanced checkout form design, additional entries, and an easy, intuitive user friendly form entry experience. Inputs are pro-actively error-checked appropriately for a smooth and proper customer experience. All bad (or empty) inputs are caught and indicated appropriately, etc. Use actual credit card formatted validation for Visa, Amex, etc. using Luhn formula (see link in resources area below for more info.)

**5. (10 Points) Shop Page Product Table**Create the entire product table using JavaScript. To help produce this table, expand your array of product objects by adding all the product description strings and filename strings for the images so that you can construct the "img" tags that use those filenames as the "src" within the table as needed. The table's HTML has to be valid, appropriate code.

**Section 5: Guidance and Resources**

To help you along in your solution to this final project, there are several valuable resources included below. In addition, class discussions will focus on issues related to the final project. There will be some resources given in class which may not be available to those who are not present in class.

**1. Where to start?**

It is probably best to try to do this final project in the order in which the features are listed in the "Grading Criteria" section. In other words, start by creating a logo, getting your product images, writing some html content for your home page, creating a few basic CSS styles to give your site a color scheme to go along with your logo, etc. No need for JavaScript quite yet.

**2. Navigation Links**

To get the navigation working, you need to know how to work with a "select" list and how to redirect the browser to a new web page. "Select list" examples appear in some of our class downloads from Canvas (payment options). Also, here is a good example of using the "location" property to direct the browser to specific page. Check out the code for this one: <http://www.faculty.rcc.edu/mcleod/js/class_demos/forms/form_demos_menu.html>

**3. Image Rollovers**

Images can be swapped out for another image using JavaScript. It is usually triggered by a mouse over (or mouse out) on that same image. Any images that are going to be swapped in should be "pre-loaded" into the browser to avoid delays in downloading the needed image and possibly ruining the timing of the swap effect. The following link discusses 2 techniques for creating this effect – one uses CSS only, one uses JavaScript only. You need to use the JavaScript method. The article and example can be found at the following link:  
<https://www.dougv.com/2009/06/the-simplest-ways-to-do-image-rollover-effects-javascript-and-css-only/>

**4. From a form page to its response page**

This technique will be presented in class and will be available in the class downloads area in Canvas.

**5. Managing the Cart Using Local Storage or Session Storage**

Local Storage: <https://www.w3schools.com/jsref/prop_win_localstorage.asp>

Session Storage: <https://www.w3schools.com/jsref/prop_win_sessionstorage.asp>

This technique will also be presented in class and will be available in the class downloads area in Canvas. Note that session storage is cleared when the browser tab is closed; local storage remains available to future sessions so the values will not be cleared unless you clear them. Either one can be used but you should keep these differnces in mind.

**6. Credit card validation (extra credit)**

For more information on validating credit card numbers (optional) using Luhn formula, see the link below:

<https://www.freeformatter.com/credit-card-number-generator-validator.html>