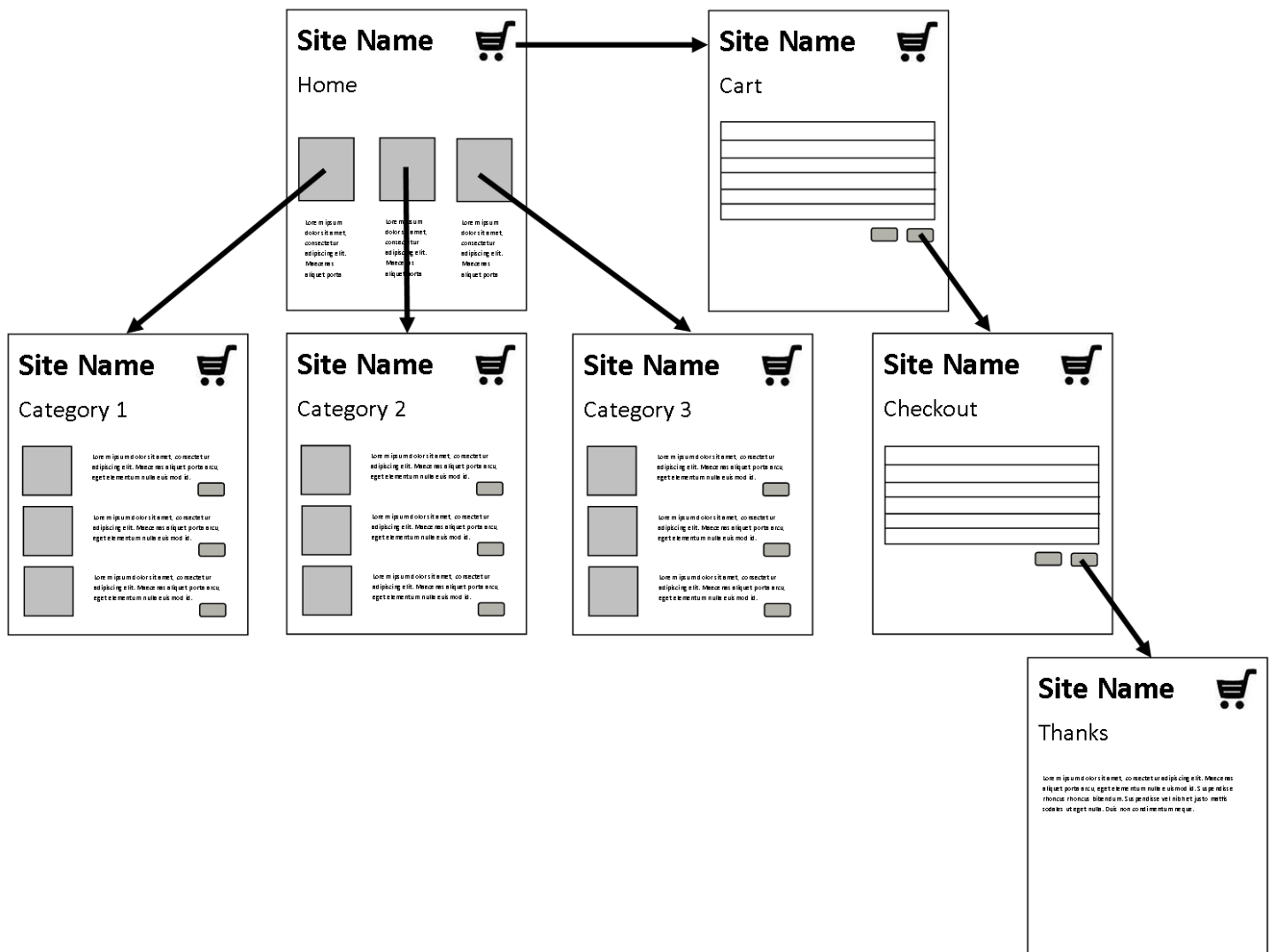


Project Description:

Create an online store with working shopping cart. On this project, you will demonstrate your ability to integrate JavaScript into a web pages as well as your ability to modify JavaScript to suit specific needs. JavaScript will include cookies and form validation.

The code for the JavaScript shopping cart has been provided in a way similar to code you might get from an online JavaScript source. Installation instructions and documentation should give you an experience similar to a professional web developer.

The instructions that follow describe how you will need to edit the template files to make a working site. By the time you finish, you should have a seven-page site that sells a minimum of 15 items in three categories.

Site Map:

Instructions:

Complete the steps in order. Periodically you will be asked to preview and test your page. If the page does not work exactly as intended, then you should not move on from that section. If directions at the end of a section state that you can move on if not complete, then you should undo any non-working code before you do so.

Understand that on this project you are not writing large blocks of HTML or JavaScript from scratch. If you find yourself wanting to do more than that, then you are probably misunderstanding the instructions.

Part 1 – Template Setup

1. Copy the template files to "Projects/10 Shopping Cart"
2. Connect each of the HTML pages to the included style.css file.
3. On each HTML page, edit the h1 tags to reflect the name of your store. Do not delete the link to index.html from the h1 tags. Clicking the store name on any page of your site should return you to the home page.
4. On each HTML page, edit the title tags to reflect the name of your store. Make sure that title tags also indicate the specific page of the site. On category pages, make sure it gives the actual category of item and not Category 1, Category 2, etc.
5. Edit the footer on each page so that it states "Site designed by Your Name".

Preview each page now. Make sure that the CSS has been applied. Also check that the title and store name are correctly displayed on all pages.

DO NOT MOVE ON UNLESS EVERYTHING TO THIS POINT WORKS CORRECTLY.

6. Now look at the home page (index.html). Look for the category div tags. Inside, there are h3 tags. Edit the Category Name to reflect the categories of items you will be selling on your site.
7. Next, make two more copies of category_1.html. Rename all three category pages so that the file names reflect the category. The file names should be in all lowercase letters and should not contain spaces. (For example, mens_clothes.html and shoes.html would be appropriate file names for category pages.)
8. Open each category page. Edit the "Shop for things in..." h2 so that it reflects the category for that page.
9. Edit the link text and the href attributes for each category link in the navigation div to reflect the renamed category files. You will need to do this for EVERY page in the template.
10. On the home page, you will also need to edit the href attributes for the links on h3 tags that link to category pages and for the images that link to each category page.

Preview your site again. Carefully, check to make sure that every link on every page works as intended. Also make sure that all generic "Category" text has been replaced with the names of the actual categories of items that you will be selling.

DO NOT MOVE ON UNLESS EVERYTHING TO THIS POINT WORKS CORRECTLY.

Part 2 – Item Pictures and Descriptions

11. For each item you intend to sell, you will need a picture. Your site should sell a minimum of 15 items and at least 4 items should be sold within each category. (It is not necessary to sell the same number of items within each category.) Gather pictures for each item and save them in the images folder.
12. The CSS will automatically scale the images, but they need to be square to display properly. Crop each image so that it is square. (The CSS makes all product images 240x240. If your images are smaller than this, they will become a little blurry when scaled. If your images are much larger, then resizing them will speed up page load times. Anything in the 200x200 to 300x300 range should be fine for your product images.)
13. On each category page, you will need to edit content within the item divs. Edit the `img src` attributes and alt text for each image. You will need to add more item divs on every page that contains more than three items.
14. Edit the item name (`h3`), item price (`h4`), and item description (`p`) for each item.

Preview each category page. Make sure that all images load and are scaled properly. Also check that the product names and descriptions display properly.

15. The home page also contains product pictures. Within each category column, edit the `img src` and alt attributes so that one image that it displays one product that is representative of each category.

Preview your home page and confirm that the images display correctly. Clicking images should take you to the corresponding category pages.

(The slideshow div on the home page can be dealt with later.)

DO NOT MOVE ON UNLESS EVERYTHING TO THIS POINT WORKS CORRECTLY.

Part 3 – JavaScript Cart Configuration

The cart javascript, css, and instructions are located in the `demo_cart` folder. Instructions are contained within the `READ_ME.txt` file.

16. Follow steps 1-4 in the `READ_ME` exactly as written. For step 5, you should put the cart div tags on the `cart.html` page within the content section. A comment indicates exactly where to put the div tags.

Preview your cart page here to confirm that the JavaScript is successfully connected to your site. Do not move on unless you have confirmed that your cart works as described in the `READ_ME` file.

DO NOT MOVE ON UNLESS EVERYTHING TO THIS POINT WORKS CORRECTLY.

17. Complete step 6 in the `READ_ME` file. You probably should get one category page working perfectly before you attempt to edit the other two. If it is not exactly clear what you are to do from the directions, then study the working cart examples within the `cart_demo` folder.
18. Step 7 will apply to our `thanks.html` page. Step 8 is completely optional. You should probably wait until the rest of your site is complete before you decide if you want to edit the cart CSS.
19. In the "More options" section, you should follow directions for option C. I have put a comment in the masthead of each HTML page that indicates exactly where the widget code should be placed.

At this point, the cart should be completely functional. Test by adding different items to the cart. You should notice that the item count in the widget area increase each time an item is added to the cart. View the cart page. Make sure that the items in the cart exactly match the items you tried to add to the cart. If the items in the cart do not match those you have added, then you may have made an error with item numbers passed to the `addToCart()` function not corresponding to the item numbers used when you edited the items in the `cart.js` configuration section. Check the button's onclick event handlers. Take your time with testing here. You need to check that every item gets added correctly. Confirm that the prices displayed in the cart match those displayed on each page. You should probably also have a friend test your cart to confirm that you aren't overlooking any errors.

DO NOT MOVE ON UNLESS EVERYTHING TO THIS POINT WORKS CORRECTLY.

Part 4 – Form Validation

20. The `checkout.html` page contains the form requesting shipping and billing information. When the checkout form is submitted, it should take the user to the `thanks.html` page but it does not yet. Give the form an action that takes the page to `thanks.html`.
21. The states dropdown menu is also incomplete. Complete the states dropdown menu so that it contains an alphabetical list of all states. The value for each option should be that state's two-letter postal code.

Preview the form and test the submit button. If it does not take you to the `thanks.html` page, then you have not assigned the form action correctly. There is no validation yet. The form should submit even if empty.

DO NOT MOVE ON UNLESS EVERYTHING TO THIS POINT WORKS CORRECTLY.

22. To get the form to validate, you will need to use the `validation.js` file provided. In the head of `checkout.html` add a script reference to the `validation.js` file.

Preview again. If you submit now, it should not submit unless the name, address, city, state, zip, email, and phone fields are filled out correctly. If the form does submit, there is either an error in your form or you have not included the `validation.js` correctly.

23. The credit card fields for card number, expiration date, and security code still do not validate. Validation functions for each are included in the `validation.js` file, but they are not being used yet. The `check()` function in `validate` is responsible for getting form data and passing it to the various validation functions.

Study the `check` function and figure out how this is done. You'll need to look really closely at what parts represent function names and which parts represent the names of form fields.

Add the required code blocks to `check()` so that the credit card fields are also validated. You should try just one field at a time in so if you make an error it will be easier to determine where it is. This is the only place you should be editing the `validation.js` file!

Preview and confirm that all form fields validate. Credit card numbers must be 16 digits long, expiration dates are entered in MM/YY form, and security codes are 3-digit numbers.

If you cannot get the credit card fields to validate, then delete any code you added so that the other information still validates. It is okay to move on even if you cannot get credit card validation working.

Part 5 - Order Confirmation

24. Once orders are submitted, the visitor needs to receive a confirmation message. On the thanks.html page, write a polite message thanking the customer for their order. Provide an order number (you can make this up) and include details about when their order will arrive. Tell them that a confirmation email has been sent and suggest that they print this page for their records. Look up sample order confirmation letters if you are unsure of wording.

At this point, your store should be fully functional. Try it out!

Part 6 – Finishing Touches

25. The home page has a slideshow div. If you created a slideshow earlier this year, you can try to integrate another one into your shopping cart site. If not, you should create a static image that fills the div entirely. The slideshow or image could tell about your store, advertise a current sale, highlight top products, etc. Be creative here. Any images included in this section should be scaled to fit nicely.

Part 7 - Style

26. The site has been deliberately styled using only black and white with simple fonts. Edit the style.css file so that the style relates to the kinds of products you are selling. At minimum, you should select colors and font-families for headings and for links. You must also demonstrate use of background colors or images. There are plenty of css classes already assigned to HTML elements on each page that correspond to sections of the style.css file. It should not be required for you to create/assign your own classes. Study the HTML and CSS carefully before you start styling.

Do not change the layout of the site unless you really know what you are doing. I will not help if you break the layout attempting to write CSS. If you want to style the shopping cart table, you can edit the cart.css file. You're on your own here too. Every table element has been assigned a class to serve as a styling hook already. It should be pretty clear when reading the CSS file what parts of the table are targeted by different classes. Do not attempt to edit the cart.js file.

Part 8 - Final Review

Go on a slow, very deliberate “shopping trip”. Make sure there are no bugs in your store. Test that every item can get added to the cart correctly, test every link, and test validation of all form fields. Make sure your cart is emptied after completing an order. Also proofread all item descriptions and your thank you message. You should then evaluate your site against the grading rubric and make any necessary corrections prior to submitting for grading. I highly suggest having a classmate test your site as well and check it against the rubric.

Part 9 – Project Submission and grading

On the day you would like your project graded, preview the home page in Komodo Edit. (Remember, our site won't be fully functional in Chrome or IE because of security restrictions on local cookies.) Close all source code tabs. Leave a rubric at your desk.

If you are not happy with your final project grade, then you may make corrections. However all regrading will be done after school. You cannot simply leave your project open for me to regrade. You must stay after and show me all corrections that you made.