

Final Assessment

In this final assessment, you'll be writing HTML and CSS. You won't need to write any JavaScript and all images will be provided for you.

Please feel free to reference your notes, books, or even research on the Internet. But please make sure all code is written by you alone and is original. In other words, please do not copy/paste any code and don't collaborate with other developers or designers whether online or in person.

The assessment should take about two hours for a developer who is new to HTML and CSS.

In this assessment, we'll be creating two pages from scratch. The web app that these pages might be part of is an online responsive shopping app that can be viewed on a phone/tablet, or laptop/desktop. Your instructor will provide you with a few screenshots to help you visualize how the final product might look but we **strongly** recommend that you read through all of the instructions to make sure you capture all of the requirements (as well as some hints that will help you along the way). You do not need to follow the steps in the order below. If you get stuck on a step, feel free to move to a later one and come back if you have time.

Setting up

1. Create a folder for the root of your web app anywhere you like, even the desktop is fine.
2. Copy the assets folder into your new root folder. All the image files you need will be found there. If you want to add more images to the folder and use them, that's fine.
3. In the root folder (not in assets), create two new files, site.css and template.html.

Setting up the template

4. Start by picking out a font you like. Anything within reason is fine as long as it is sans-serif and not Comic Sans. In your site.css file, set this font as the default font for all of your pages.
5. Edit template.html. Add the four required parts to make it a proper HTML page.
6. Add header, footer, and main semantic tags. While you're creating this template, you won't put anything in the <main> because this page will become the basis for all your other pages later.

Create the header

7. Using CSS in your site.css file, give the header a background color of white.
8. Add a logo image to the top left. (Hint: `` for a start) Using CSS in your site.css file, size that image to about 50 pixels by 50 pixels.
9. Add a list of links to the header. Please look at the screenshots to see what they should say. The links won't really go anywhere. In fact, make their href property be "#". This will allow the links to be clickable, but won't take any real action. Style them to be side-by-side (hint: float or flex) with no bullets (hint: list-style-type) as in the screenshot. Make them black and not underlined (hint: text-decoration).
10. Bonus! Only if you are ahead of schedule, add a down arrow symbol to the right of all of these links to represent a dropdown menu. See if you can add them in the CSS stylesheet only. (hint: You'll use `::after {content: '\25bc';}`).
11. Add a form to the header with an input box. It should have a placeholder that says "Search". The input box should have rounded corners and plenty of padding.
12. Bonus! It'll be really pretty if you add this property to your text box when the user focuses: `{ box-shadow: inset 0px 0px 10px red; outline: none;}` If you have time, test that out.
13. Make sure the form has a submit button that says "Search". Note that the form doesn't have to actually work. You are not being asked to implement any kind of real search functionality, just the visual part.
14. Add a hamburger menu symbol (hint: `≡`) to the right of the Search. This also doesn't have to work, just look nice.
15. Make both the hamburger menu and the search form right-justified. (Hint: One solution would be to add a `` between the left-justified things and the right-justified things and give it all of the flex-grow weight.)
16. This may be tough to test at this point, but make your header always visible at the top of the window/viewport so that no matter where the user scrolls, they never lose the menu.
 - Hint #1: When you make the position fixed, it may scrunch up horizontally. When it does, you can fix that by going `"top:0; left:0; right:0"` which will stretch it out again.
 - Hint #2: Later, after you put some scrollable content on the page you can test it. When you do, you'll see that the top content may be hidden behind your fixed menu. To fix that, put some padding at the top of the `<body>`.

Create the footer

17. In the template's footer section, Create three sections. These will be for:
 - Customer service
 - Company stuff

- Connect with us

Each section has a list of links in it. Just as before, their href properties should point to "#". Refer to the screenshots to see the text values each list should have. Note that all of these sections have a similar look. This is a clue that they should share a style. (Hint: one way is to use classes)

18. Now style the sections and lists. In particular notice that none of the lists have bullets and there is plenty of space between the list items. The links are not underlined (hint: use the ":link" pseudo-class and text-decoration property). The sections themselves have a background color and there is space between the sections (hint: margin) and around the contents of each section (hint: padding).
19. Underneath all of those, add a copyright notice with your (fake) company name and this year's date. Go ahead and center the notice and make the text lighter and smaller.

Making the template responsive

Remember that we want the pages to look good on any size screen. In the next few steps you'll be making your pages respond to the screen width using media queries. If you've made your HTML too rigid above, this is a good time to refactor them so they can be responsive.

20. Edit your site.css file and create a section at the bottom for your @media query. This is where you'll add CSS code to make the next few steps happen.
21. The header should no longer be locked to the top of the viewport on a smaller screen. You should be able to scroll it out of view. (Hint: display: relative)
22. Also in the header, your logo should be above the menu choices which are above the search form.
23. The hamburger menu should disappear altogether on a smaller screen. (Hint: display: none).
24. your logo should remain small on a bigger screen but be much larger, say 200px by 200px on a smaller screen.
25. In the footer, the three sections should be one on top of another on a small screen rather than side-by-side on a larger screen.
26. Go through your template one more time and make any last-minute adjustments you'd like. If you want to make changes from this point forward, you'll need to make it in three places because we're about to clone the template for our other two pages.

Creating the index page

As you're going through these, don't forget that you have screenshots that will help you to visualize what is being described.

27. Copy this template to an index.html page. Make sure all of the following steps are only done on your index.html page and not your template.

28. Edit index.html. In the main, add a Featured Item section. This will be where we display a product that we're trying to promote.
29. Make sure Featured Item section has two sub-sections, product picture on the left and product info on the right.
30. In the product picture section you should have an image of the item.
31. In the product info section/div/whatever to the right, make sure it has an item name, price, quantity input, and an add to cart button.
32. Add another major section below all that for catalog categories.
33. That section should have four categories. Each has an image with the description overlaid. (Hint: Whenever you want things to overlap, you'll use absolute positioning and set the top or bottom and the left or right).

Creating the product page

34. Copy the template into a new file called product.html. This file will be for viewing a single product, like when the user clicks on an item from the catalog and wants to see its details.
35. Make sure its main section has two sub-sections, product pictures on the left and product detail info on the right.
36. In the product pictures section you should have a big hero image of the item with three smaller images in a row below. (Hint: Put them in a grid with two rows and three columns. Make the hero image's max-width to be 100%. Make the smaller images' max-width to be, like, 100px.)
37. The details on the right should include a product name, product description, price, quantity box, and an add to cart button. Style these things as shown in the screen shot.
38. There should be four pictures, one big one and three smaller ones below in a horizontal row.

Making the pages responsive

Now that you've created these two pages, let's say you needed to make them look good on a phone or tablet also. We're going to make the pages responsive. As you code, test with either the developer tools or just decrease the width of your browser.

39. In index.html on a small screen, the categories should be vertical instead of horizontal. The featured item section should be omitted.
40. In the main section of product.html on a small screen, the product name should be first, followed by the main product picture. The other three images should be omitted. The remaining product info should appear last.
41. Test your site out one last time to make sure it meets the requirements and works at all screen widths.

Submitting the site

42. Please email these files to your instructor for evaluation.
43. If you haven't moved the assets folder or added any other supporting files, you should be able to just email the three html files and one css file to the

email address he or she has provided to you. If you have altered it in some way or you just want to play it safe, zip the entire site folder and email it.

Thank you for participating in class this week!