

## WEB222 Assignment 5

### Overview

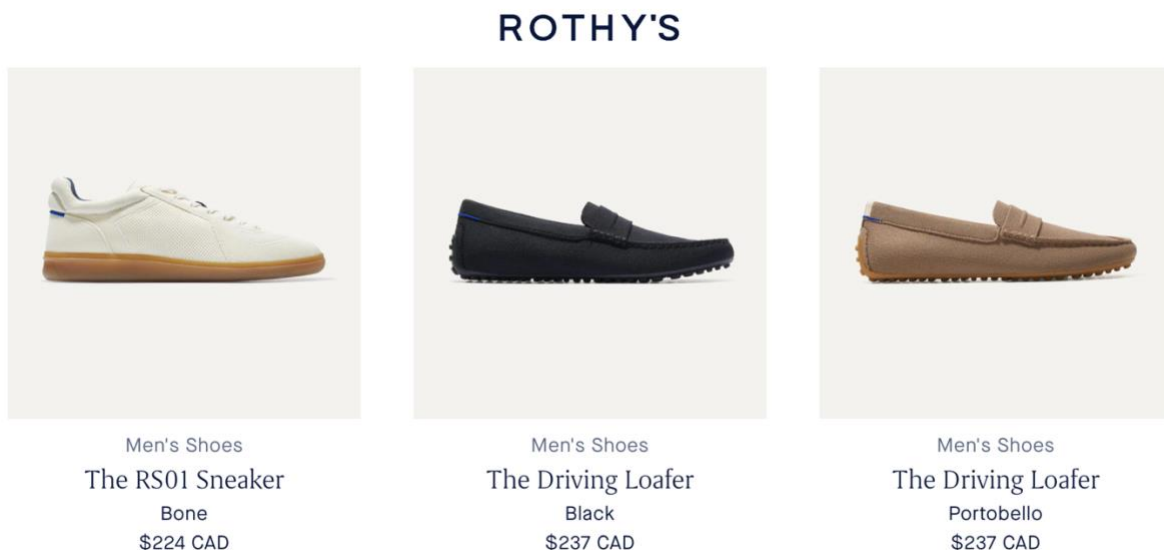
This assignment is designed to have you practice building more complex HTML and CSS layouts. We will continue to iterate on your previous Assignment 4 web store's *static* and *dynamic* web content.





You are asked to update the design of your fictional online store. This time, instead of displaying your products in an HTML table, you will create visual product “cards” that show a picture, name, description, and price.


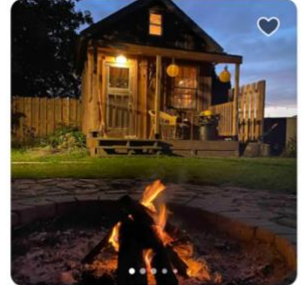


You must do all the work for this assignment on your own. You may consult your notes, use the web for inspiration, but you should not copy code directly from other sites, or other students. If you need help, ask your professor.

### Cards

Cards on the web, much like trading or playing cards, are rectangular areas that allow you to visually present a lot of related data. We often see them used in online stores, social media, and anywhere that we want to mix images, titles, and text in a rectangle. Here are some real-world examples from Rothy's, Amazon, and airbnb:



<p><b>Bestseller</b></p>  <p>BIC Xtra-Sparkle Mechanical Pencil, Medium Point (0.7 mm), 24-Count (Pack of 1), Refillable Design for Long-Lasting Use,...</p> <p>★★★★☆ ~ 27,826</p> <p>\$11<sup>25</sup> (\$0.47/count)</p> <p>Save more with Subscribe &amp; Save</p> <p>✓prime FREE One-Day</p> <p>Get it <b>Tomorrow, Mar 21</b></p> <p>More buying choices</p> <p>\$5.46 (7 new offers)</p>	 <p>Staedtler 13246CB18P11TH The Premium Pencil, Norica, HB2 special lead, 12 Plus BONUS 6 for Total 18 Count</p> <p>★★★★☆ ~ 329</p> <p>\$10<sup>49</sup> (\$0.87/count) <del>\$13.25</del></p> <p>✓prime FREE One-Day</p> <p>Get it <b>Tomorrow, Mar 21</b></p> <p>Only 9 left in stock.</p>	 <p>Staedtler Norica 50 Graphite HB2 Pencils +50 eraser caps +1 Sharpener</p> <p>★★★★★ ~ 450</p> <p>\$19<sup>48</sup> <del>\$24.00</del></p> <p>FREE Delivery for Prime members</p> <p>More buying choices</p> <p>\$18.97 (19 new offers)</p>	<p><b>Bestseller</b></p>  <p>AmazonBasics Pre-sharpened Wood Cased #2 HB Pencils, 150 Pack</p> <p>★★★★★ ~ 43,227</p> <p>\$17<sup>09</sup> (\$0.11/count)</p> <p>\$16.24 with Subscribe &amp; Save discount</p> <p>✓prime FREE delivery by <b>Tuesday, Mar 22</b></p>
---	--	---	---

 <p>Mansfield, Ontario 77 kilometres away</p>	 <p>Lincoln, Ontario 60 kilometres away</p>	 <p>Horning's Mills, Ont... 81 kilometres away</p>	 <p>Creemore, Ontario 93 kilometres away</p>
\$183 CAD / night Mar. 28 – Apr. 4	\$113 CAD / night Apr. 29 – May 6	\$132 CAD / night Apr. 10 – 17	\$328 CAD / night Sep. 26 – Oct. 3

There are lots of resources you can use to learn more about creating a card, for example:

- [https://developer.mozilla.org/en-US/docs/Web/CSS/Layout\\_cookbook/Card](https://developer.mozilla.org/en-US/docs/Web/CSS/Layout_cookbook/Card)
- [https://www.w3schools.com/howto/howto\\_css\\_cards.asp](https://www.w3schools.com/howto/howto_css_cards.asp)
- <https://www.freecodecamp.org/news/learn-css-basics-by-building-a-card-component/>

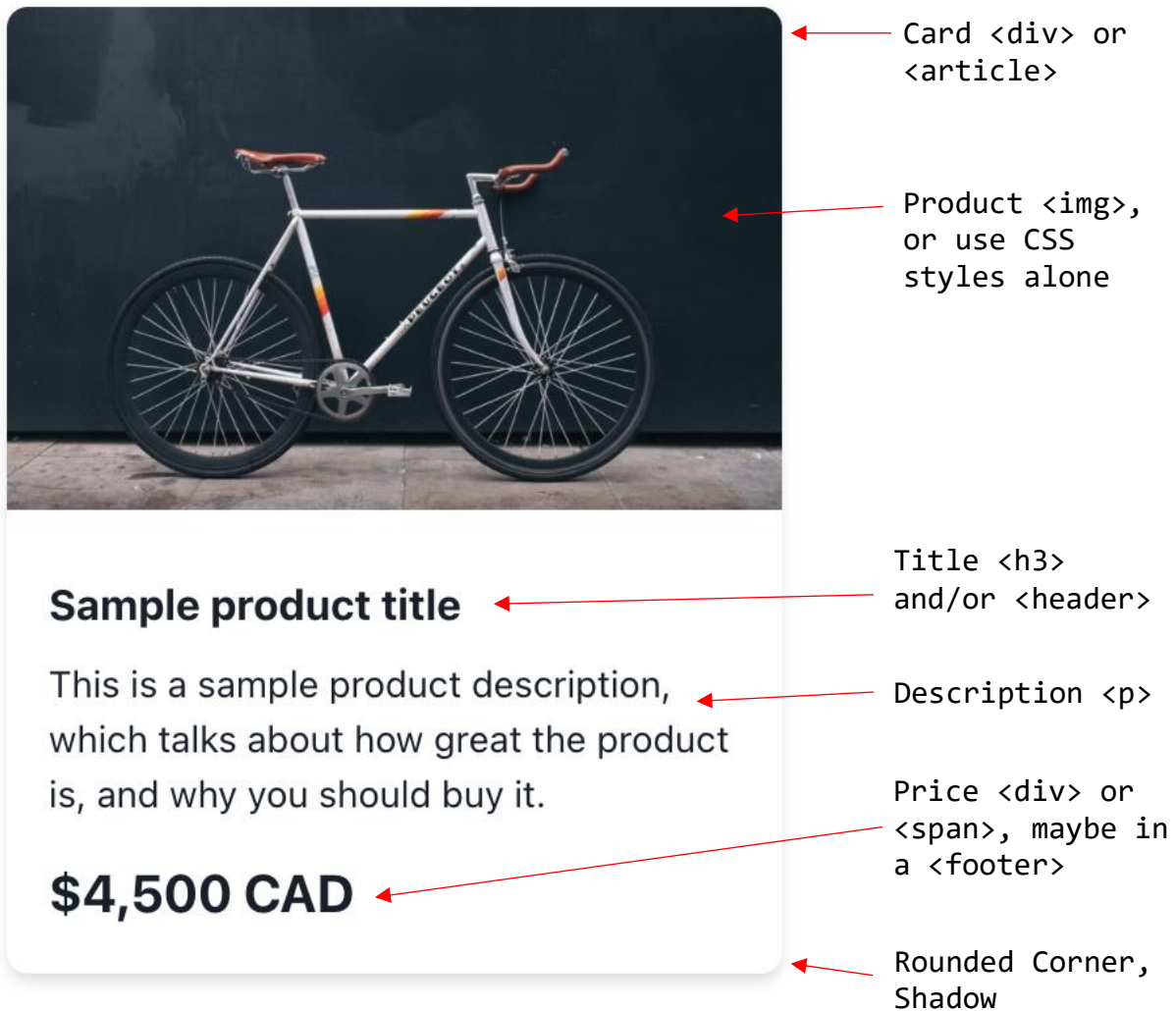
## Update Your Store to Use Cards

Modify your solution to Assignment 4 in order to **replace the HTML table with rows of cards**. To do this, you should follow these steps:

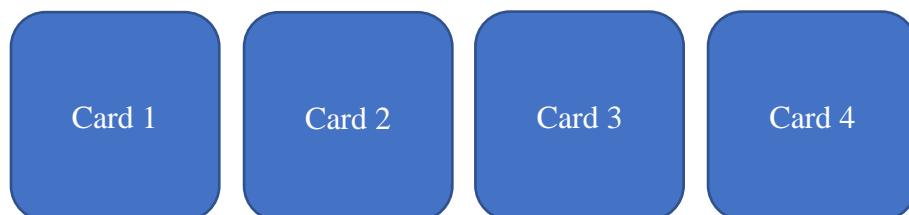
1. Create a copy of your Assignment 4 project, so you don't lose your previous work.
2. Start simple. In your HTML file, create a single product card (i.e., a <div>) that includes an <img> of the product, a heading (e.g., <h2> or <h3>) for the name, a <p> for the

description, and a `<span>` for the price (you can modify the HTML elements you use, these are just suggestions).

Use CSS classes on your card's elements in order to apply colours, fonts, margins, padding, borders, etc. until you have something that you like. Here's an example, which uses rounded corners, a subtle shadow, different font sizes, and a large photo at the top.



3. Create rows of cards. Use **Flexbox** or **CSS Grid** to create rows that repeat your cards across and down. For now, you can copy and paste your card from step 1 over and over in order to repeat it. Make your page look good with rows of 3 or 4 cards. Adjust the spacing, size, etc. until you're happy with how it looks.



4. Find **at least** 3 product images to use in your cards. You don't have to find an image for every single product (i.e., you can repeat the same ones), but you should have at least 3 unique images.

Make sure you optimize the images so they are not too big to download (i.e., don't use a 5000x6000 image in a card that uses 400x200).

You can use <https://squoosh.app/> for images that you download. Or you can also use a trick with <https://unsplash.com/> images to resize them automatically via the URL. For example, the bike above is <https://unsplash.com/photos/tG36rvCeqng>. Here's the full-sized image <https://images.unsplash.com/photo-1485965120184-e220f721d03e> (it's 3.8M in size, and 4440x2960). We can reduce that image by adding some parameters to the image URL: `?auto=format&fit=crop&w=750&q=80` to crop and resize it to 750 pixels wide, and reduce the quality a bit to 80%, like this: <https://images.unsplash.com/photo-1485965120184-e220f721d03e?auto=format&fit=crop&w=750&q=80> See <https://unsplash.com/documentation#dynamically-resizable-images> for more details.

5. Update your `src/products.js` file so that each Product Object has an `imageUrl` property. This imageUrl should be the URL or filename of an image to use for this product in your card. Multiple products can use the same image URL if you don't have enough unique images for each product.
6. Remove the card's HTML you wrote earlier and write a function in JavaScript that creates them dynamically instead. Your function should accept a product Object from your products array and create a card in the DOM, using the HTML and class names you wrote above. Here's some code to get you started:

```
function createProductCard(product) {  
  // Create a <div> to hold the card  
  const card = document.createElement('div');  
  // Add the .card class to the <div>  
  card.classList.add("card");  
  
  // Create a product image, use the .card-image class  
  const productImage = document.createElement('img');  
  productImage.src = product.imageUrl;  
  productImage.classList.add("card-image");  
  card.appendChild(productImage);  
  
  // ... rest of your card building code here  
  
  // Return the card's <div> element to the caller  
  return card;  
}
```

}

7. Modify your page load and button click events so that instead of creating table rows for each product, you call your `createProductCard()` function and get back the card's `<div>...</div>` element, which you can now place in the DOM (e.g., using `appendChild()`). Clicking your category buttons should show the correct product cards on the page.

### Coding:

Use your copy of the website starter project in the assignment-4 ZIP file. Install all dependencies by running the following command in the root of the assignment (e.g., in the same directory as `package.json`):

**`npm install`**

Your code should all be placed in the `src/` directory.

### Running a Web Server:

You can start a local web server to test your code in a browser by running the following command:

**`npm start`**

This will start a server on <http://localhost:8080>, which you can open in your web browser

To stop the server, use CTRL + C

### Submission:

When you are finished, run the following command to create your submission ZIP file:

**`npm run prepare-submission`**

This will generate `submission.zip`, which you can hand in on Blackboard.