Assignment 5

Instructions

You are asked to create an imaginary online store. Your site will focus on the HTML, CSS, and JavaScript for the site's front-end, but not require a back-end.

Please see detailed requirements in the comments within `src/index.html`.

To submit your assignment, you must successfully run the `npm run prepare-submission` build script discussed below. Please read all of the following information carefully, and don't skip any steps.

Design Your Store

What will your store sell? In the past students have focused on clothing, jewelry, second-hand clothing, make-up, cakes, tea, and computers. Your store can sell whatever you want, and you can use this as a way to say something about your personality and interests. It can be fun, funny, or serious.

Ideally, you'll build something in this assignment that you'll be proud to show in interviews when applying for co-op positions. Former students have used their stores to showcase their web development skills and get hired.

Get Inspired

You probably already use many online stores. Take some time to go visit the ones you like, and explore some new ones to get inspired. Here are some examples of stores to get you started:

- Shoes at https://www.nike.com/ca/ and https://www.goat.com/
- Stickers at https://www.stickermule.com/ca
- Computers at https://www.canadacomputers.com/, https://www.bestbuy.ca/en-ca, and https://www.apple.com/ca/
- Men's Clothes at https://www.frankandoak.com/ and https://shop.tees.ca/
- Women's Clothes at https://www.aritzia.com/en/default and https://6bygeebeauty.com/
- Fitness equipment at https://www.roguecanada.ca/
- Food at https://www.pipsnacks.com/
- Art at https://www.nickmayerart.com/
- Coffee at https://www.deathwishcoffee.com/
- Records at https://www.funkymooserecords.ca/

Notice what's similar and different about each of these, how they use fonts, colours, and images. Pay attention to common layouts. Stores all sell different things, but they use many common patterns in their design and user interface. Try to identify a few of the effective techniques you see in the stores you like and use it to guide your own design.

NOTE: you need to create your own store design, and shouldn't copy anything directly from another site, but it's OK to get inspired by existing designs.

Setup

Install all the necessary dependencies using the npm command (installed with node):

```
npm install
```

This should download and save all the necessary files to a folder named node_modules in the current directory.

If you have trouble getting "npm install" to work:

- Did you install node.js?
- If you type "node --version" in your terminal, does it print the version?
- Are you in the right directory (you must cd into the correct directory)

If you need help, ask your classmates or talk to your professor.

Writing your code

All of your HTML, CSS, and JavaScript files should be placed in `src/`.

The build scripts discussed below all assume your files are located there.

Using Build Scripts

There are a number of build scripts you can run to help you check, validate, and use your files. All of the scripts are run using the following command format

"npm run script-name":

1. To start a web server so you can browse all the files in `src/*`:

   ```
   npm run server
   ```

   You can now open the URL http://localhost:3000. Use CTRL+C to stop the server. The web page will automatically reload when you save your files.

2. To run all the tests to validate your HTML, CSS, and JS:

   ```
   npm test
   ```

   You can also run each check on its own:

   ```
   npm run validate
   npm run eslint
   npm run stylelint
   ```

3. To properly auto-format your HTML, CSS, and JavaScript using https://prettier.io:

   ```
   npm run prettier
   ```

   This will reformat and rewrite all the .html, .css, and .js files

4. To check your JavaScript files for errors using https://eslint.org/:

```
npm run eslint
```

If you're not sure what an error means, look at the rule name in https://eslint.org/docs/rules/

5. To check your CSS files for errors using https://stylelint.io/:

   ```
   npm run stylelint
   ```

   If you're not sure what an error means, look at the rule name in https://stylelint.io/user-guide/rules/

6. To check your HTML for errors using http://html5.validator.nu:

   ```
   npm run validate
   ```

7. To run steps 2-6 above one after another, and prepare your code for submission:

   ```
   npm run prepare-submission
   ```

   You can now upload the submission.zip file to Blackboard.