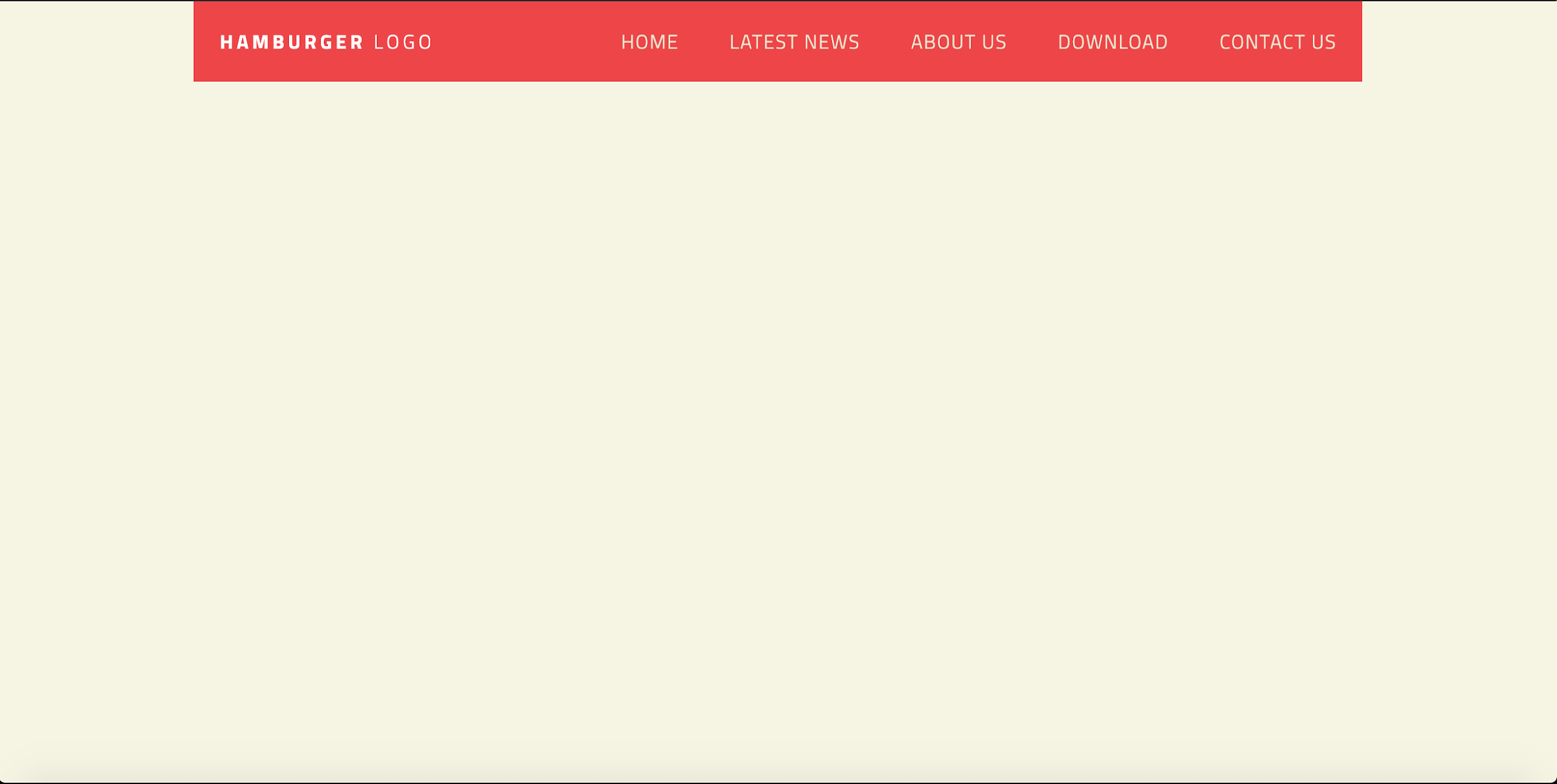
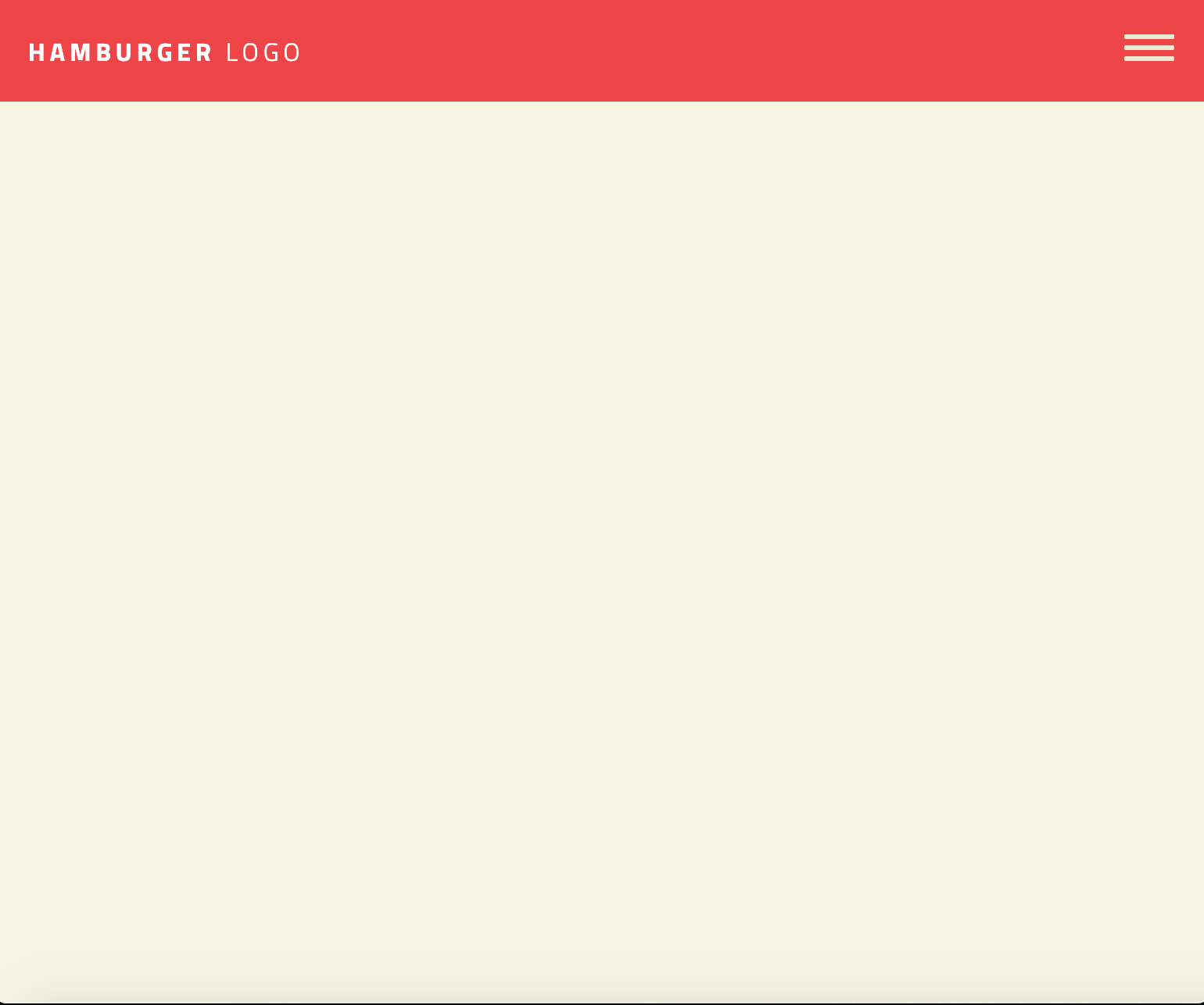
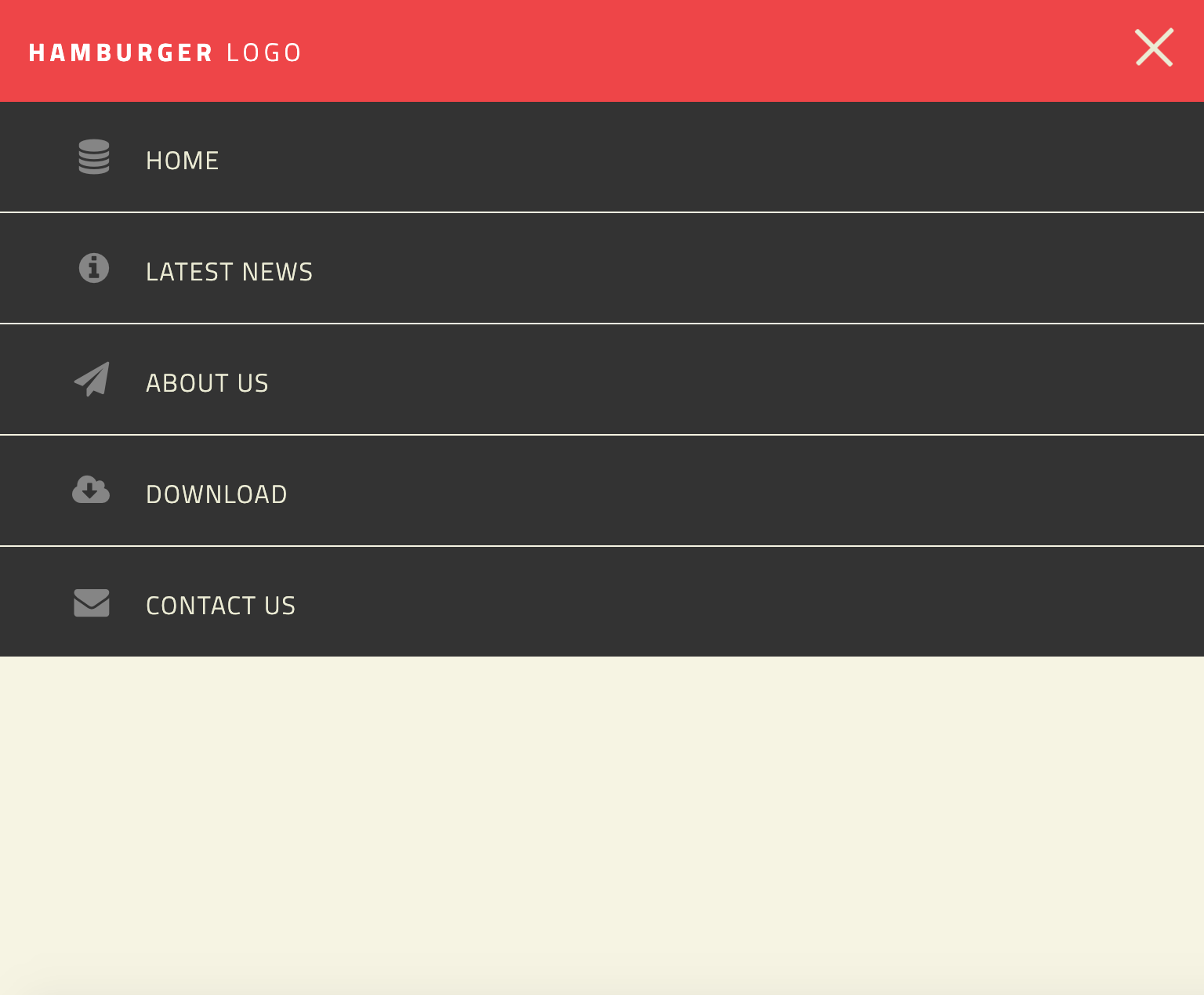
Hamburger Menu JS Assignment

AIM Code School

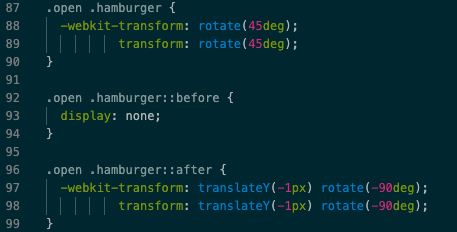
Prepared by Vanessa Kasun







# Let’s Begin…

1. Create a project folder named **hamburger-menu-js**
   1. Create **css** folder
   2. Create **images** folder
   3. Create a **js** folder
      1. Inside the js folder create a file called **index.js**
   4. Create **index.html** with required tags
2. Inside the **<body>** tag create a **<div>** element with a class of ***wrapper***
   1. Inside this element create a **<nav>** tag with the class of ***site-nav***
   2. Every element on this page will have a **box-sizing** property with the value ***border-box.***
   3. The general **background color** for this example site it **white**
   4. The **font** is **‘Titillium Web’** with the ***letter-spacing*** property and a value of **1px.**
   5. The ***wrapper*** has a **width** of 100%, a **maximum width** of **960px** and the **margin** needs to be set so the page starts with no spaces around the content.
   6. The **<nav>** element will have the **margin** and **padding** properties, both values will be set to **zero**.
3. Directly inside the **<nav>** element, create an **<h1>** tag with the class ***logo***
   1. The **content** for this tag will be the word **Hamburger**. Before closing the tag, after the word Hamburger…
      1. Create a **<span>** tag with the content word **Logo**.
      2. Close the **<span>** tag and then finally…
      3. Close the **<h1>** tag
   2. **Please Style the *Logo* class using your css knowledge**
4. Still inside the **<nav>**, create a **<div>** with the class **menu-toggle**
   1. Inside the **menu-toggle** **div** create another **<div>** with the class of ***hamburger*** (this will be our icon for the mobile view)
      1. The position property will be set to absolute
   2. **Please Style the rest of the *menu-toggle* class with your css knowledge**
   3. **.logo span** needs to have a **font weight** of 400.
   4. Using the **Comma Combinator Selector,** in css, select the ***hamburger*** class while also creating two **pseudo-elements** for it. One for **::before** and one for **::after.** 
      1. **The CSS for these selectors are as follows:**
         1. Content: ‘ ’;
         2. Display: block;
         3. Background: #ebebd3;
         4. Height: 3px;
         5. Width: 2em;
         6. Border-radius: 3px;
      2. WE WILL NEED TO COME BACK TO THESE SELECTORS TO ADD OUR TRANSITION PROPERTY WITH THE CORRESPONDING VALUES SO OUR TRANSFORMATION CAN TAKE PLACE WITHIN OUR LOGO THAT WE JUST CREATED USING THE PSEUDO ELEMENTS.
   5. NEXT, Still in **CSS**, select **.hamburger::before**
      1. **Transform** the **translateY** value to be (-7px);
      2. Also, please add your **-webkit-transform** version as well.
   6. REPEAT the same step for **.hamburger::after**. EXCEPT the values will be:
      1. (4px) instead of (-7px)
5. Even though we do NOT have the class of ***open*** in our HTML, we are going to go ahead and create one in CSS with a subset of **pseudo elements** in order to apply it to the **hamburger menu** using jQuery for Mobile Responsive Web Design. So add this to your css after the last selector : (these act as the actual hamburger menu icon transformations when it’s clicked. )
   1. 
6. DON’T FORGET TO GO BACK TO **STEP 4(d)** SELECTORS AND ADD THE **TRANSITIONS** PROPERTY! (***VALUES: ALL, EASE-IN-OUT, 350MS***)Let’s switch back to HTML. STILL inside the **<nav>** tag…
   1. Create an **<ul>** with the class of ***open desktop***
   2. Each list item **<li>** will consist of an **<a>** tag with the ***href*** attribute.
   3. In addition to that, it will also contain an **icon** from Font Awesome Here are the icons you will need:
      1. Database
      2. Info circle
      3. Paper plane
      4. Cloud download
      5. Envelope
7. Make sure to **close** the unordered list, the nav, and the wrapper <div>
8. In your **head** tag, where the **metadata** goes, don’t forget to link the **css**, **google font, font awesome, and the latest version of jquery,** which can be found [here](https://www.w3schools.com/jquery/jquery_get_started.asp).
9. ALSO, even though we have not yet coded anything in the JS file, please link it using this syntax:
   1. **<script src=“js/index.js></script>**
10. Let’s finish up our CSS before we move onto the jQuery. We have left to style the **<ul>** the **<li>** the **<a>** tags inside the **<nav>.** This requires a bit more explanation…
    1. First, we need to make the **nav ul** a f**lex container**. Since we are focusing on the mobile styles first, the **flex-direction** needs to have a value of **column**.
    2. We have the icons set up to act the list style. So in CSS the **list-style-type** value needs to be **none**.
    3. Make the correct margin and padding adjustments
11. Earlier I said we were focusing on mobile view first. Even though we are coding this to be desktop AND mobile friendly, for this project and for most mobile responsive ones, coding the mobile CSS first is popular among developers. Because that is the case, and we are currently focusing on our <nav> navigation, <ul> unordered list, elements we need to make sure that this “version” of the navigation menu doesn’t display by default. Why? Because the goal is to see the menu as a desktop view by default, and only when the user screen is a certain size, does the mobile view take over and override the desktop CSS properties.
    1. So with that being said go ahead and add the **display:none;** property to the ***nav ul*** selector as well.
12. Next, select ***nav ul.opening.*** This will act as the **display** we want for when the user **clicks** on the hamburger menu. Notice we don’t have this exact class within our **HTML** anywhere. That is because we will be using **jQuery** to apply this class to the hamburger menu once the user **clicks** on it. So what that means is that we are able to create **Classes** within **CSS** that we can apply to an **HTML** element with **JavaScript** or **jQuery**.
    1. Looking at the example images, we can see that we want to **display** this as a **block** element. For this selector you will also need to add the height property with the value you think is correct for the desired outcome.
13. Go ahead and put a bottom border on each of the **nav** **li** elements. The bottom border needs to be:
    1. 1px
    2. Solid
    3. #f6f4e2
       1. In addition to that, you will also need to make sure there is **no border** on the very last **<li>** element within the **nav**. ( **HINT:** this requires you to use a **pseudo class** in conjunction with the **nav li** selector. )
14. Now for the navigation anchor tags! This selector I am going to leave all to you. Use your best knowledge to accomplish the desired look.
    1. Don’t forget to add the **nav a :hover** and **:focus** selectors in addition to your nav anchor tag as well. The **background color** for that will be **#E4B363**
15. The last step we are going to take before coding our desktop CSS, is to style the ***.site-nav--icon*** class.
    1. Change the font size
    2. The right margin
    3. width
    4. Text align
    5. And the color of rgba(255,255,255,.4)

CSS Media Query Desktop Overrides..

1. Let’s start by adding our @media Rule. The conditions will be min-width of 780px.
2. The first CSS selector following this rule will be the ***menu-toggle*** class. Since we are now focusing on the desktop view version of this website, we don’t want this part of the HTML to show. This is the class that houses the hamburger menu. So we need to set this **display** property to none.
3. Well continue by making changes to the **flex** property and the **border** to the navigation list items. We will no longer need the **border** so we need to use the same property and value as a we did in **step 13(i)**
   1. We will change the flex value to read:
      1. **flex: 1 1 auto** (The alignment abilities or auto margins can be used to align flex items along the main axis. flex: auto;. This is equivalent to flex: 1 1 auto)
4. Next, you will needs to make changes to the **nav list item**, **anchor tags** the properties you will need to override:
   1. Padding
   2. Margin
   3. Border
   4. Background-color: #f04343;
      1. Change the background color to rgba(255,255,255,.10); when the mouse :hovers and :focuses
5. The **logo** class needs to have a **margin** of zero, a **padding-left** of 20px, and the **flex** property with the value of 0 0 200px;
6. Select **nav** and make the necessary changes in order to **display** the menu as you see in the image. Use your knowledge! We’ve done this lots of times!
7. Finally, we want to make sure the ***site-nav--icon*** doesn’t display on the desktop view so use the correct property and value to achieve that.

# For the jQuery…

1. Wait until your document is ready. Then…
   1. When the user clicks on the menu-toggle class, the unordered list needs the ‘opening’ class toggled on and off.
   2. The ‘open’ class needs to toggle on and off of **THIS** selected element.

HOORAY! Take a look in your browser and you should see the desktop view. Once you shrink your screen or change it to mobile view in chrome inspect, you will see the mobile css take effect!

If you have any errors, double check your code. Re-read through the instructions and make sure you read them word for word.