Development Strategy

For this project, you will be writing most of your code in **js/app.js**. Note that it's very important that you *plan your project* before you start writing any code! Break your project down into *small* pieces of work and strategize your approach to each one. With these bite-sized amounts, it'll be easier to debug and fix any issues that appear.

Feel free to implement your own design workflow, but if you get stuck -- here is a walkthrough to get you up and running!

1. **Start by linking your app.js**. where should this file go based on your present knowledge? We'll test some other locations later.
2. **Build out your HTML and at least 3 content sections**. The rest of your functionality relies on these sections.
    - Take a quick look at all the HTML elements in **index.html**. Note the values for their `id`, `class`, and `data` attributes. To manipulate the DOM, you'll be using many of the tools and methods you've learned on these elements (and on those that you will create).
        - For a refresher on the data attribute, visit here.
    - Which data structure can you use to store these sections? This data structure can represent all sections for your page, so it might be a good idea to save it to a variable.
    - How you would iterate (i.e., loop) over this data structure?
    - Think about how you can create, say, an *unordered list* (i.e., bulleted list) in HTML from this structure, and where you be placing that list.
    - Think about how you'll test whether a section is in the viewport.
    - What actions are you performing that will cause interactivity with the DOM?
3. **Build the navigation menu**. This will dynamically create a navigation menu based on the sections of the page. This can be a particularly useful trick when you begin working with content management systems or APIs when you are uncertain of where the items will be.
    - Are you listening for an event for the navigation to build?
    - Where are you placing the navigation?
    - Where is the text for each navigation item coming from and where are you anchoring to?

- How are you going to add each navigation item to your menu? (Hint: there are several ways to do this. Do some research to figure out which makes the most sense for your situation. Performance? Clarity?).

4. **Add functionality to distinguish the section in view**. While navigating through the page, the section that is active in the viewport/closest to the top should be distinguished from the other sections.
    - Are you listening for an event for sections to become active?
    - How are you going to test which section should be highlighted?
    - How can we use `classList` methods to change the CSS being displayed? What about removing that CSS?
    - Check the HTML and CSS files to ensure that what you chose is updated in the other locations.

5. **Add the functionality to scroll to sections**. Clicking on a navigation item should scroll to the appropriate section of the page.
    - Which event are you listening for (hint: you were just reading it)?
    - There is a default event occurring that we need to stop. How?
    - If you don't recall how HTML page anchors work, read more to figure out which variables you should set.
    - There are several javascript methods for scrolling. Which seems like it may be the most simple?

6. **REFACTOR**. At this point, your code should be working properly. Ideally, refactoring happens while you are developing, but as a new developer, you often don't have the whole picture in your head to be able to do so properly. Let's clean the project up.
    - Have you run your code through a linter? We request you still follow Udacity standards when the code is complete, but running it through an eslinter is going to help you get started in refactoring.
    - Are you using ES6 const and let?
    - Are all your functions using ES6 arrow functions?
    - Is your code DRY? Are there any pieces that would be better served as a helper function to avoid duplication?
    - How is your code structured? Have you created functions for the main functionality with properly scoped variables? Starting out it's likely that you will have a globally scoped variables on occasion until you learn more about closures

and design patterns. But placing your code into functions is a great way to make your code more readable and a way to avoid globally scoped variables.
    - Are you using the best method for your iterations?

7. **Add additional sections to your HTML document**. See how the navigation builds.
8. **Test the performance**. The performance of your page can be affected by how you write your javascript as well as where you load your javascript.
    - Test loading the javascript in the head vs at the end of the body. What issues arise? Is there a way to still load in the head without breaking the page? What is the performance like compared to loading at the end of the body?

9. **Suggested**:
    - Add an active state to your navigation items when a section is in the viewport.
    - Hide fixed navigation bar while not scrolling (it should still be present on page load).
        - Hint: setTimeout can be used to check when the user is no longer scrolling.
    - Add a scroll to top button on the page that's only visible when the user scrolls below the fold of the page.
    - Update/change the design/content.
    - Make sections collapsible.

Version Control

Although not a requirement, we recommend using Git from the very beginning. Make sure to commit often and to use well-formatted commit messages that conform to our Git Style Guide.

Udacity Style Guides

You should write your code and markup to meet the specifications provided in these style guides:

- CSS Style Guide
- HTML Style Guide
- JavaScript Style Guide
- Git Style Guide

Still Not Sure How to Begin?

To reiterate, be sure that you are comfortable with the content from JavaScript and the DOM. After all, this entire project is about DOM manipulation!

A note on plagiarism: Viewing someone else's code to get a general idea of implementation, then putting it away and starting to write your own code from scratch is okay. **Please do not copy someone's code**, in whole or in part. For further details, check out this guide regarding plagiarism.