Here is what you will need to complete the assignment:

1. (If you haven't already) Create a GitHub.com account and a repository that you will use for this class.
2. (If you haven't already) Follow the Development Setup Video (beginning of Module 1) instructions on how to create a repository and set it up such that you can host and view your finished web pages on GitHub Pages, i.e., GitHub.io domain name. You will need to provide that URL for your peer review.
3. Create a folder in your repository that will serve as a container folder for your solution to this assignment. You can call it whatever you want. For example, module3-solution or mod3\_solution, etc. Create an index.html file inside the solution container folder, e.g., module3-solution/index.html.
4. The implementation of the page you will be creating should follow the mockup illustrations shown below. You are provided 3 mockups: desktop and tablet (same), mobile, and mobile with mobile menu dropdown shown. Your implementation has to be JUST 1 page. In other words, you will be creating a single, responsive page.
5. Your page must include a CSS file. No inline styles allowed. Your CSS file should be placed into a css folder under the solution container folder, e.g., module3-solution/css.
6. For this assignment, you are to use Twitter Bootstrap CSS Framework as much as possible. I suggest you start with copying the starter bootstrap files and folders we discussed in Lecture 25 part 2. If you've cloned/downloaded the code example repository, it should be in the examples/Lecture25 folder. Copy the contents of examples/Lecture25 to your solution container folder (e.g., module3-solution) as a starting point..
7. Since we are using Bootstrap for this assignment, instead of specifying pixel ranges, I will define our desktop, tablet, and mobile views in terms of Bootstrap CSS class prefixes, i.e., md, sm, and xs.

* Desktop mockup illustration should correspond to Bootstrap md-based classes
* Tablet mockup illustration should correspond to Bootstrap sm-based classes
* Mobile mockup illustration should correspond to Bootstrap xs-based classes

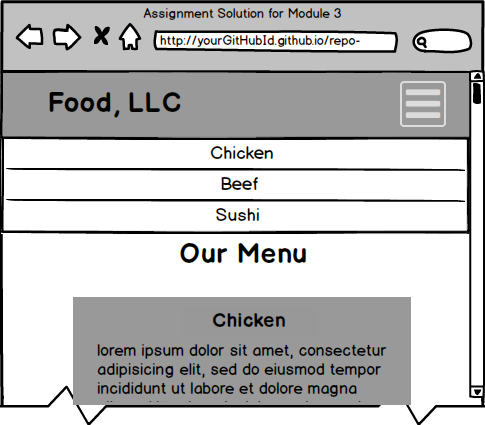
1. Navbar: Create a navbar that scrolls away together with the page (the navbar should become invisible and is not fixed to the top when you scroll the page down). The navbar should have a company name (i.e., navbar-brand class) called "Food, LLC" that is aligned to the left side of the navbar. (*See* <https://getbootstrap.com/docs/3.3/components/#navbar>. *Make the browser window narrower to see the mobile menu button appear in the first example shown at the provided link.*)  
     
   For desktop and tablet view, the navbar should not contain anything else. No other buttons should be visible. (*Hint: use 'visible-xs' class.*)
2. Navbar - Mobile View: Create a simple menu button (3 horizontal bars). When the user clicks that button, a dropdown menu should appear (as illustrated in Mobile Open Menu illustration below.) The dropdown menu should contain 3 items: Chicken, Beef, and Sushi.  
     
   The dropdown menu should take up the entire width of the browser window. Make sure the dropdown menu has a background color set that distinguishes it from the rest of the content.  
     
   (*Hint: See* [*https://getbootstrap.com/docs/3.3/components/#navbar*](https://getbootstrap.com/docs/3.3/components/#navbar) *and Lecture 31 for examples on how to accomplish this.*)
3. Page Heading. The page heading that says Our Menu should be centered within the browser window. You must use a Bootstrap class to accomplish this.  
     
   (*Hint: look for a Bootstrap class that centers text, see* [*https://getbootstrap.com/docs/3.3/css/#type-alignment*](https://getbootstrap.com/docs/3.3/css/#type-alignment)*.*)
4. Create a single really tall section that will use the Bootstrap Grid and take up the entire width of the browser window (minus some margins, of course) for all views: desktop, tablet, and mobile. To make the section really tall, you can either fill it out with a LOT of text or simply set its height to something like 1000px. It needs to be tall enough to cause scrolling down to be required to view the bottom of the section. Make sure its background color is set to distinguish it from the rest of the content. (*Hint: don't forget to have an element with a class='container' or class='container-fluid' wrapping your grid. Remember that to have the grid do something "always", i.e., no matter what browser window size, use the col-xs-... classes. In this case, since we want the section to take up the entire row, use col-xs-12.*)

Both the tablet view and the desktop view of what's graded and required is the same. Here is the mockup illustration of the desktop & tablet version of the site (only required graded parts shown):

[](https://github.com/jhu-ep-coursera/fullstack-course4/blob/master/assignments/assignment3/images/desktop.png)

Here is the mockup illustration of the mobile view (only required graded parts are shown):

[](https://github.com/jhu-ep-coursera/fullstack-course4/blob/master/assignments/assignment3/images/menu-collapsed.png)

[](https://github.com/jhu-ep-coursera/fullstack-course4/blob/master/assignments/assignment3/images/mobile-expanded.png)

**\*\*\* REQUIRED GRADED ASSIGNMENT ENDS HERE \*\*\***