**Responsive Web Design Projects - Build a Technical Documentation Page**

* **Objective:** Build a [CodePen.io](https://codepen.io) app that is functionally similar to this: <https://codepen.io/freeCodeCamp/full/NdrKKL>.
* Fulfill the below [user stories](https://en.wikipedia.org/wiki/User_story) and get all of the tests to pass. Give it your own personal style.
* You can use HTML, JavaScript, and CSS to complete this project. Plain CSS is recommended because that is what the lessons have covered so far and you should get some practice with plain CSS. You can use Bootstrap or SASS if you choose. Additional technologies (just for example jQuery, React, Angular, or Vue) are not recommended for this project, and using them is at your own risk. Other projects will give you a chance to work with different technology stacks like React. We will accept and try to fix all issue reports that use the suggested technology stack for this project. Happy coding!
* **User Story #1:** I can see a main element with a corresponding id="main-doc", which contains the page's main content (technical documentation).
* **User Story #2:** Within the #main-doc element, I can see several section elements, each with a class of main-section. There should be a minimum of 5.
* **User Story #3:** The first element within each .main-section should be a header element which contains text that describes the topic of that section.
* **User Story #4:** Each section element with the class of main-section should also have an id that corresponds with the text of each header contained within it. Any spaces should be replaced with underscores (e.g. The section that contains the header "Javascript and Java" should have a corresponding id="Javascript\_and\_Java").
* **User Story #5:** The .main-section elements should contain at least 10 p elements total (not each).
* **User Story #6:** The .main-section elements should contain at least 5 code elements total (not each).
* **User Story #7:** The .main-section elements should contain at least 5 li items total (not each).
* **User Story #8:** I can see a nav element with a corresponding id="navbar".
* **User Story #9:** The navbar element should contain one header element which contains text that describes the topic of the technical documentation.
* **User Story #10:** Additionally, the navbar should contain link (a) elements with the class of nav-link. There should be one for every element with the class main-section.
* **User Story #11:** The header element in the navbar must come before any link (a) elements in the navbar.
* **User Story #12:** Each element with the class of nav-link should contain text that corresponds to the header text within each section (e.g. if you have a "Hello world" section/header, your navbar should have an element which contains the text "Hello world").
* **User Story #13:** When I click on a navbar element, the page should navigate to the corresponding section of the main-doc element (e.g. If I click on a nav-link element that contains the text "Hello world", the page navigates to a section element that has that id and contains the corresponding header.
* **User Story #14:** On regular sized devices (laptops, desktops), the element with id="navbar" should be shown on the left side of the screen and should always be visible to the user.
* **User Story #15:** My Technical Documentation page should use at least one media query.

LIST OF HTTP CODES