Landing Page

Build Navigation Menu Dynamically With Javascript

itida egyptfwd initiative UDACITY

Tip 1

Selection of all the sections via
document.querySelectorAll('section');

Tip 2

Create a Document Fragment via **document.createDocumentFragment();** to append created nav items to it for the sake of performance.

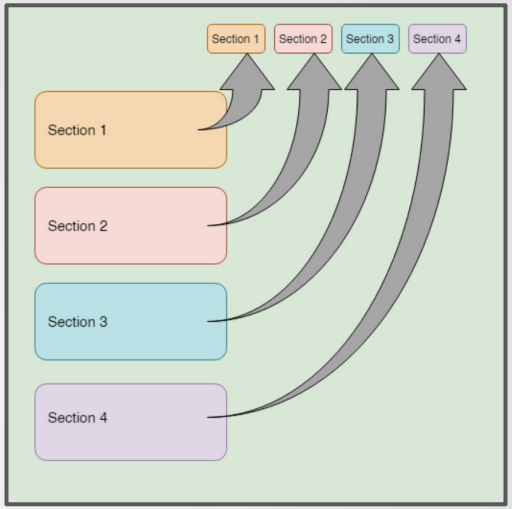
itida egyptfwd initiative UDACITY

Tip 3

Loop on the sections list created in TIP #1 using a for.. Of loop or any other type **of loop** and create **<li>** elements with each iteration via **document.createElement('li');** and append anchor <a> element as a child to it. Use the value of each section's **data-nav** attribute to create the text inside each anchor **<a>** tag, and the value of each section **id** attribute to create the value of anchor's **href** attribute in addition to symbol #.

Tip 4

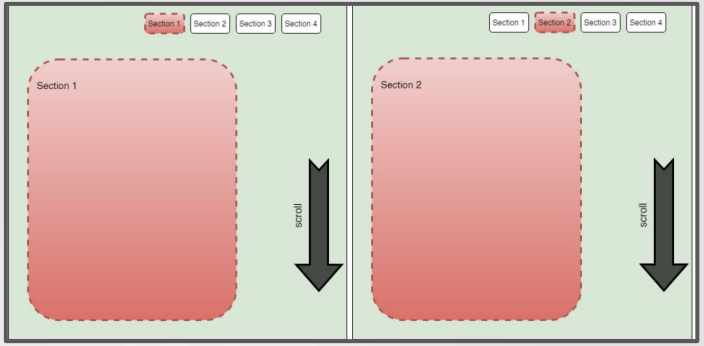
Append each nav item to the previously created Document Fragment via **docFargment.appendChild('listItem');** . And after closing the loop append the filled Document Fragment to the unordered list <ul> element via **list.appendChild('docFargment');**

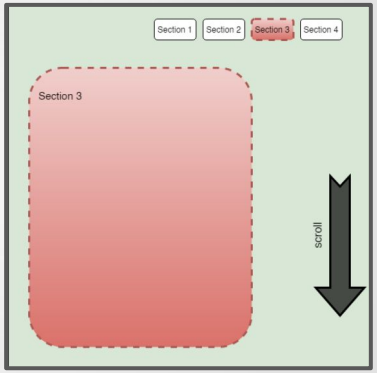


itida egyptfwd initiative UDACITY

Tip 5

Use a scroll event listener
window.addEventListener('scroll', toggleActiveState);
Inside toggleActiveState function use **intersection observer** to detect the section inside the viewport (on screen) while scrolling.
You can learn about **intersection observer** syntax and how to use it from this tutorial. And see a practical example with code snippet here.





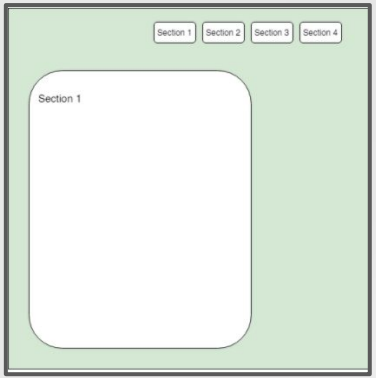
Tip 6

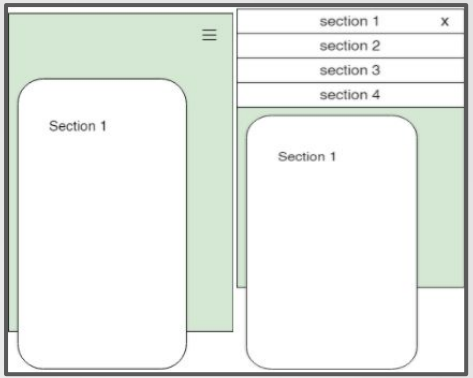
Inside the observer, when the section is intersecting (on screen) we should remove the active CSS classes (highlighting style classes for both sections and nav items) from all the sections, and all the links first by looping on all the sections and all the nav items and removing the classes from each one of them via **section.classList.remove('your-active-class');** and **navItem.classList.remove('item-active-class');** for the sections and nav items respectively. Then add the same active classes for the intersection section and its related nav item.

itida egyptfwd initiative UDACITY

Tip 7

Use a click event listener on the nav menu links
navLink.addEventListener('click', scrollToSection);
Inside scrollToSection function:
1. Prevent default action of the clicking on a link (which is jumping right to the section) via **event.preventDefault();** where **event** is an object comes with the event listener and should be added firstly as an argument to the **scrollToSection** function when declaring it like this function **scrollToSection(event){}**
2. Detect the selected section related to the clicked nav link. We can use the value of the nav link href to select it.





Tip 8

Also inside scrollToSection function call **scrollIntoView()** method on the selected section above like this: **selectedSection.scrollIntoView({behavior: “smooth”, block: “center”);** Where the block is the vertical alignment of the section when scrolling.

Tip 9

For a responsive nav menu as above images for smaller screens use CSS media query and inside it put the code for converting the nav menu from the horizontal one to the vertical one.
You can learn how to do it from this this tutorial.

Tip 10

The readme file is a file written in markdown language and is used to introduce, and explain a project, and give instruction on how to use it, etc.

In our project we just have to explain what the project is about, and what we have used to create it.

See this example of a readme file