**Exercise 3 – Accessible navigation**

**Headings**

In the index.html file, in order to achieve appropriate nested heading hierarchy,

1. Big headings in 3 sections i.e. *News & events, Studies, Researches*: <div> was changed to <h1>.
2. Small headings in section i.e. *Global temperature record, At iam decimum annum in spelunca iacet, Quo plebiscito decreta a senatu est consuli quaestio Cn，Advance your scientific education， Join our renowned research groups，Latest publications*: <div> was changed to <h2>

**Article and their titles**

In the index.html file, div elements were switched to HTML5 sematic tag. This allows us to express the structure and content of the webpage more clearly, making the page more readable, accessible, and semantic.

1. Replaced the **<div>** tag that represents the entire "News & events" section with a **<section>** tag. Because **<section>** tag is semantic to represent the "News & events" section.
2. Replaced the outer **<div>** tag of each article with an **<article>** tag. Because **<article>** tag is semantic to represent an individual article.
3. Replaced the **<div>** tags for the three article titles with **<h2>** tags. Semantic heading tags, specifically **<h2>** tags, were used to identify the article titles.

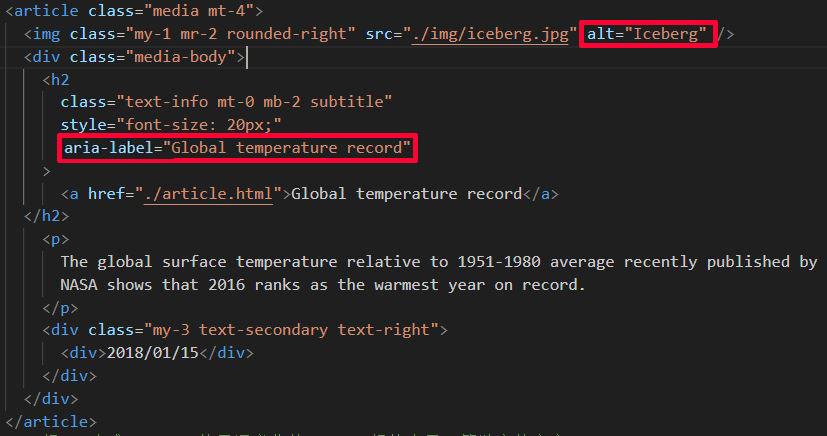
In the index.html file, in the previous code, screen readers would read the content of each element sequentially, including images, text, and dates. Screen readers typically follow the default reading order, which is from top to bottom and from left to right. In order to improve the user experience, it is desirable for the screen reader to announce the article title when the user is positioned within the article content.

1. The **aria-label** attribute was added to provide an additional text description for the article title so that screen readers can read the article title.

2. At the same time, we also provided a proper description for the image by adding the **alt** attribute.

This way, when screen reader users are positioned within the article content, the screen reader will first announce the article title, followed by the body content of the article. This improvement enhances the user experience by allowing users to understand the current article title without navigating to the heading.

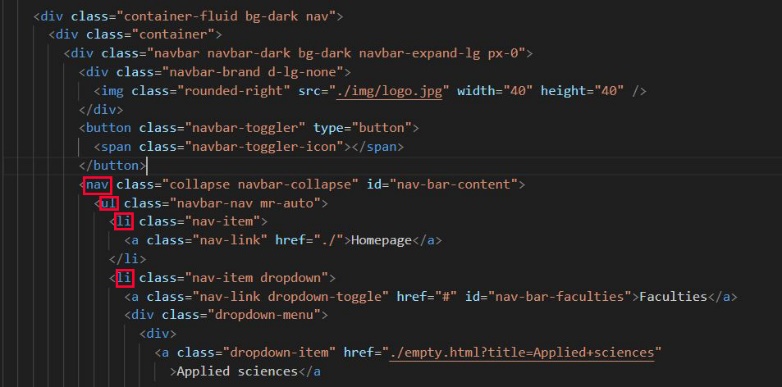
Take the first article as example:



**Menu structure**

In the index.html file, generic **<div>** elements can be used to build menu structures, but for expressing the hierarchy of menus and the structure of submenus, it is more appropriate to use semantic HTML elements such as **<nav>**, **<ul>**, and **<li>**. This improves code readability and makes the menu structure clearer and more understandable.。

1. <div class="collapse navbar-collapse" id="nav-bar-content"> was changed to <nav class="collapse navbar-collapse" id="nav-bar-content">：From a semantic perspective, **<nav>** is more suitable than **<div>** for representing the semantic meaning of a navigation bar.
2. <div class="navbar-nav"> was changed to <ul class="navbar-nav">：Navigation menus are typically represented as an unordered list (**<ul>** element). It is more appropriate to use the **<ul>** element to represent the structure of a navigation menu.
3. <div class="nav-item"> was changed to <li class="nav-item">, <di Each menu item in the navigation menu should be a list item (**<li>** element). It is more appropriate to use the **<li>** element to represent each item in the navigation menu, as it aligns with the structure of a menu.



**Drop-down menu**

In the original website, it was found that a screen reader user won’t be updated about the change of status, i.e. the sub-menu isn’t displayed. Therefore, the menu was repaired so that the screenreader will read "menu", "expanded/collapsed" correctly. In the index.js file,

1. In the "Add event listeners" section, set the **aria-expanded** attribute of all dropdown menus to "false" to ensure that the submenus are initially closed.
2. In the ‘**toggleDropdown’** function, toggle the expansion state of the menu items based on the current aria-expanded attribute value. Use the ‘**classList.toggle’** method to add or remove the show class of the submenu, thereby displaying or hiding the submenu.
3. In the `**handleKeyDown**` function, when the "Escape" key is pressed or the "Space" key is pressed while the menu item is expanded, call the `**toggleDropdown**` function to toggle the expansion state. Use the `**classList.toggle**` method to add or remove the `show` class of the submenu, thereby displaying or hiding the submenu.。

With the above modifications, when a menu item is clicked, the submenu will visually expand or collapse. The screen reader users will be updated about the expansion state of the submenu by setting the `**aria-expanded**` attribute and adding/removing the `**show**` class.

**Menu keyboard interaction**

In the index.js file,

1. In the `**toggleDropdown**` function, the expansion and collapse of the dropdown menu are achieved by setting the `aria-expanded` attribute and adding/removing the `show` class.
2. In the `**handleKeyDown**` function, keyboard events are listened to. When the "Escape" key or "Space" key is pressed and the dropdown menu is already expanded, the `**toggleDropdown**` function is called to toggle the expansion state. This is achieved by using the `**classList.toggle**` method to add or remove the `show` class on the submenu, thereby displaying or hiding the submenu.
3. In the **window** object, a **keydown** event is listened to. When the "Tab" key is pressed, all dropdown menus are iterated through. It checks if the event target is within the range of menu items and if the menu item is expanded. If these conditions are met, the respective dropdown menu is closed.

With the aforementioned modifications, when the user presses the "ESC" key, any expanded submenu will automatically close, and the focus will return to the menu title. Additionally, by listening to the "SPACE" key, users can now activate menu items using the "SPACE" key, providing an improved user experience. Lastly, by monitoring the "TAB" key, any open submenus will be closed when the focus moves away from the menu.

**Skip links**  
In the index.html file, for the three articles, three different shortcuts were respectively added for screen reader users to quickly jump to a section of the page.

Taking the second article as an example, its corresponding codes are: <a href="https://hivedata.com/at-iam-decimum-annum-in-spelunca-iacet/">At iam decimum annum in spelunca iacet</a>