

## CPSC 2030 Assignment 3: CSS Grid and Basic jQuery

Total Marks: 10

### Setup:

- Download the starter files from D2L.
- HTML code, and some CSS code, has been provided. You will write your styles in a file called **mystyles.css**. You may edit the HTML files only where specifically indicated (for Tasks 3 and 4). Do not edit the files **style.css** and **search.css**!
- **There are some screenshots and a demo video in the assignment folder that will show you what the site should look like when you're done. Study these carefully!**

### Task 1: Desktop Layout

- Create a new file called **mystyles.css** and write your styles in that file.
- The mobile layout for this assignment is already done for you. Basically, I've just set up some simple colors, fonts, background images, and a one-column layout. In **mystyles.css**, use a **min-width media query** to specify a desktop layout for **1000px** and wider. You will use Grid to create the entire layout for the page. **You should not use the float property anywhere in your CSS!**
- The **.sticky-bar** element should use a **three-column** layout to arrange the **<h1>**, **<button>**, and **<ul>**. (The form has been taken out of the flow of the document using the position property, so you don't have to worry about placing this in the grid.) The order of the elements should be changed so that the button is on the right of the layout.
  - o The **<h1>** and the **<button>** should have columns with width **max-content**, which means they will be expanded to the full width of their content without text wrapping.
  - o The **<ul>** column should have a width of **1fr**, which will cause it to fill the space between the h1 and button.
  - o Use the **align-items** property on the **.stick-bar** element to ensure all the elements within the columns are vertically centered, instead of positioned at the top by default.
- The **<ul>** should itself be a grid with four columns, each with width **max-content**. Use the **justify-content** property to have all the menu items aligned to the right side within the grid container.
- The primary content of the site should be arranged in **two columns**. The right column should have a fixed width of **300px**, whereas the left column should grow to fill any leftover space.

## Task 2: Sticky Nav Bar

- To make the navbar (which has the **.sticky-bar** class) stick to the top of the window when scrolling, we're going to use a new **position** property value called **sticky**. You'll have to also add one additional property to make this work. (hint: it's a property that very commonly accompanies the position property.)

## Task 3: jQuery Search Form

- I've written all the styles for the search form; all you have to do is implement it using jQuery. (You can't see it because I've hidden the form using **display:none**. You'll use jQuery to hide and show it.)
- Notice that I've added an **aria-label** attribute to each button. Add some text to the value of these attributes to describe the function of the button for screen readers. This is necessary for accessibility, since icons may not be useful to users with visual disabilities.
- Get the code linking to the jQuery library from the following source and paste it into the head of your page: <https://developers.google.com/speed/libraries/>
- There is a script tag in the head section of your document. This is where your jQuery code will go. I've included the basic starter function which is activated when the DOM has loaded. (There's a comment in the code telling you where to write your own code.)
- Configure the search window to display when the search icon in the nav bar is clicked (The magnifying glass icon).
  - To do this, use the **on()** method to detect when the user clicks on the button, and use the **fadeToggle()** or **fadeIn()** method to cause the form to appear, with fade animation.
  - Using the same technique, configure the search screen to close when the user clicks on the **x** in the top right corner. You can use the **fadeOut()** method for this.
  - Make sure the search functionality is fully keyboard accessible and can be navigated using the **tab** key (use Chrome to test this):
    - When the user clicks the magnifying glass button to expand the search screen, use the **focus()** method to make sure focus is placed on the **search bar**. You should be able to immediately type content into the search bar without having to click on it.
    - When the user clicks the **x** button to close the search form, use the **focus()** method again to place focus on the magnifying glass button in the nav bar.

## Task 4: Navbar Animation

- Notice that in the demo video when the user scrolls far enough down the page for the navbar to stick to the top of the window, there is an animation in which the **border-width** of the **.sticky-bar** decreases. Your task is to implement this animation using a combination of jQuery (to add and remove classes when scrolling) and CSS (to implement the actual animations using these classes).

- When the user scrolls on the page, check the distance that the user has scrolled so far by using the `$(window).scrollTop()` function. This will return the distance from the top of the screen that the user has scrolled. Use **if** and **else** statements to either add or remove the **sticky** class depending on whether `$(window).scrollTop()` is above or below a certain distance. You can use the methods `addClass()` and `removeClass()` to achieve this. You can measure the height of the `<header>` element by using the `height()` method.

More info:

- [https://www.w3schools.com/jquery/css\\_scrolltop.asp](https://www.w3schools.com/jquery/css_scrolltop.asp)
- [https://www.w3schools.com/js/js\\_if\\_else.asp](https://www.w3schools.com/js/js_if_else.asp)
- [https://www.w3schools.com/jquery/jquery\\_css\\_classes.asp](https://www.w3schools.com/jquery/jquery_css_classes.asp)

- Use the **transition** property to animate the **border-width**.

### Hand in:

When you have completed the assignment, please delete all screenshots and demo videos, rename the working folder to **a3\_firstname\_lastname**, zip it (do not use some other kind of archive like rar or 7z), and hand it in to D2L. Following these instructions will help the marker.

### Check your Work:

- Look at the screenshots and demo videos in the assignment folder. Is there anything you've missed?
- **[3 marks]** CSS Grid layout
- **[1 marks]** Sticky nav bar
- **[3 marks]** Search form: aria-label has been set, buttons all work properly (with fade in/out animation) and focus is applied properly when expanding and hiding the form.
- **[3 marks]** Animations have been added to the sticky nav bar.

Note: up to -30% may be deducted for poor coding style and files handed in improperly.

### Resources

- Event Listeners in jQuery:  
<http://api.jquery.com/on/>
- Focus an element using jQuery:  
<https://api.jquery.com/focus/>
- Use jQuery to fade in an element that has display:none  
<http://api.jquery.com/fadein/>

### Acknowledgements and References:

- Jordan Millier, Langara College
- jQuery API <http://api.jquery.com>