CPSC 2030 Assignment 2: Flexbox Layout and CSS Animation

Total Marks: 10

Setup:

- Download the starter files from D2L.
- HTML code has been provided. You will write your styles in the file called **styles.css**. **You may not edit the HTML code! (adding links to stylesheets is fine though.)** The purpose of this assignment is for you to use advanced CSS to style the document without changing the mark up.
- **There are some screenshots and demo videos in the assignment folder that will show you what the site should look like when you're done. Study these carefully!**

Hand in:

When you have completed the assignment, please create a **zip** archive (not some other kind of archive) from your **a2_starter_files** folder and hand it in to the **Lab 2** folder on D2L. **Please do not hand in the screenshots and demo videos since it will make it difficult for the marker to download**.

Task 1: Mobile Layout

- The mobile layout will be a simple, one-column layout. Look at the screenshot and apply the necessary styles to match it as closely as possible. I'd like you to use the mobile-first approach, which means you will start by implementing the mobile layout (without using a media query) and then implement the desktop layout (using a min-width media query below the code for the mobile layout.)
    - Fonts and Colors:
        - The h1 font is **Sacramento.**
        - The h2 font is **Poiret One.**
        - The colors used are **cornflowerblue, hotpink**, and **lavenderblush** (for the background of the menu items.)
    - Layout:
        - As you can see, the nav menu has been placed at the bottom of the HTML document, beneath the main element. However, we'd like the nav menu to be displayed in the browser above the main and below the header.
            - Use **display:flex,** the **flex-direction property,** and the **order** property to achieve this change in element order.

Task 2: Desktop Layout

- Below your mobile styles, use a **min-width media query** to specify a desktop layout for **800px** and wider. You will use Flexbox to create the entire layout for the page. **You should not use the float property anywhere in your CSS!**
    - Layout:
        - The content of the page should take up **80%** of the width of the window and be horizontally centered. Set this up using the **align-items** property. (do not use margin:auto!)
        - Implement a two-column layout within the **main** element using flexbox.
        - We'd like the left column (Featured Images) to have a fixed width of **300px**; when the window is resized, the right column should either grow or shrink without the left column being affected. Set up this behavior by using the **flex-grow** and **flex-shrink** properties on the two columns.
        - Set the **height** of all images to **150px** and the **width** to **auto**.
        - For the **Image Gallery** section, use flexbox properties to match the layout in the demo video.
        - For the **Featured Images** section, use **flex-direction:column** and other properties to get the desired layout.
        - For the navigation menu, use the **flex-grow** property to get the items to perfectly fit within the width of the page.
        - Set up any remaining styles that are present in the desktop screenshot (like background colors, etc.).
        - **Optional challenge:** when you make the window quite narrow, you might see that the left column extends all the way down to the bottom of the page. Using only flexbox properties, try to get the left column to be no taller than its content.

Task 3: Animation

- Notice that in the demo video when the mouse pointer hovers over the images, they grow slightly. Implement this behaviour using the **transform** property (to change the size of the image) and the **transition** property (to animate this change).
- When the mouse pointer hovers over a menu item, the top, right, bottom and left borders appear in sequence. Implement this by using the **transition** property with multiple values.
    - **Hint:** animate the colors of the individual sides of the box. Remember that you can set the color of each side separately.
    - **Another hint:** remember that you can specify the delay for each animated property.

- Use the **@keyframes** rule to animate the background image. The image should move with constant speed to the right, and the animation should repeat indefinitely without any noticeable jumps.
    - **Hint:** Check the width, in pixels, of the background image. Note that in order for the animation to loop smoothly, the image should move to the right by a distance that is equivalent to its width. Use the **background-position** property to change the horizontal position of the background image.

Check your Work:

- Look at the screenshots and demo videos in the assignment folder. Is there anything you've missed?

- **[2 marks]** Styling: fonts, colors, background colors, margins, paddings, etc.
- **[5 marks]** Flexbox layout
- **[3 marks]** Animations

Total: 10 marks

Resources:

- Flexbox quick guide: https://css-tricks.com/snippets/css/a-guide-to-flexbox/
- Flexbox tutorial: https://flexbox.io/
- Lynda.com's Flexbox video tutorial: https://www.lynda.com/CSS-tutorials/Advanced-Responsive-Layouts-CSS-Flexbox/383780-2.html
- A cute frog-based Flexbox game: http://flexboxfroggy.com/
- Keyframes rule: https://www.w3schools.com/cssref/css3_pr_animation-keyframes.asp
- Keyframes animation advanced: https://css-tricks.com/snippets/css/keyframe-animation-syntax/
- Transition animations: https://css-tricks.com/almanac/properties/t/transition/
- Transform property: https://css-tricks.com/almanac/properties/t/transform/

Background texture by Welsley: https://www.transparenttextures.com/inspiration-geometry.html