HTML/CSS Exercise: Set the controls for the heart of the sun.

In this exercise, you will build a responsive page. You will use grid and flex, transitions, and a small amount of JavaScript.

Be sure to check what each section is worth.

Two Big Hints

- you can do transitions between **column widths in grid**
- the **overflow** property is your friend.

Starter Content: Font Awesome Icons

To make life easier for you, I have created a starter file that has the site name and two menus with font awesome icons. You will be given this file in class.

You will need to create the rest of the content in the document.

Content & Typography (10 marks)

Set all text in the document to the **System UI** font stack. Choose appropriate font-weights.

The document has 15 articles. Each article has:

- a random feature image from picsum.photos. It must be **greyscale**.
- a headline with **eight words** of unique lorem ipsum
- two paragraphs, each with **twenty-five words** of unique lorem ipsum
- a counter (see next section)

In the header, the type size scales with the browser width. At all times, it wraps onto at least two lines.

Counter

The following code will add a "counter" to your articles. You will need to style the counters, of course.
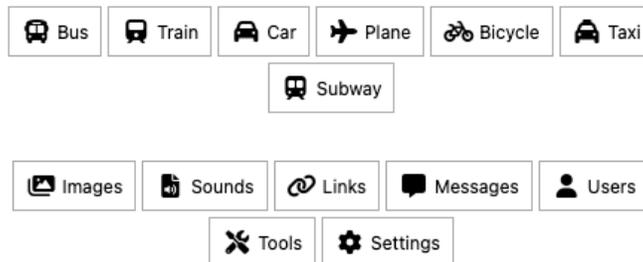
```
article {
  counter-increment: story;
}
article::before {
  content: counter(story)
}
```

Layout

Phone View (20 marks)

The site title and links are centered. The links are arranged left to right. The articles stack in a single column. Make everything look **as close as possible** to this screenshot.

Tablet View (20 marks)

- A button becomes visible in the top of the screen. Note the alignments.
- The menus appear on the left and the right.
  The text of each link is not visible. Do not use inaccessible techniques like *display:none*. Think about the hints I gave earlier.
- The articles are arranged with this repeating pattern:
  - two in one row
  - one in the next row
  - etc.

Desktop View (20 marks)

The only difference here is that the repeating pattern of articles changes to **three in one row, two in the next**, etc:

neart oꞮ tne sun

Lorem ipsum dolor sit amet consectetur adipisicing elit.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Aspernatur rem quia labore, esse sapiente tempora porro dicta voluptates ipsa animi laudantium eius quis quasi doloremque!

Nisi dolore dignissimos repudiandae doloremque accusamus fugit est, sequi neque fugiat dolor deleniti error. Dolor qui similique ratione at architecto tempora numquam praesentium officia blanditiis.

Quo alias ducimus deleniti rerum voluptate, consequatur id!

Lorem ipsum dolor sit amet consectetur adipisicing elit. Impedit facilis, distinctio aliquid reiciendis asperiores repellat perspiciatis nemo animi, explicabo architecto iure rem aperiam maxime error.

Eos soluta voluptas assumenda, quam aut ad esse nulla aliquid possimus enim cumque accusantium, hic eligendi eum accusamus perferendis explicabo vitae exercitationem repellat, illum et?

Cum neque, fuga debitis saepe quo velit quasi.

Lorem ipsum dolor sit amet consectetur adipisicing elit. Debitis dignissimos mollitia, atque animi est ipsum, praesentium fugiat velit dolorum aspernatur sint quas, dolores quasi eligendi?

Libero nulla nam repellendus excepturi necessitatibus ab voluptatem illo fugit labore ut deserunt quibusdam minus, numquam deleniti commodi quae voluptatum ea laboriosam sunt quam neque?

Behaviour

Hover Conditions (20 marks)

This applies to the tablet and desktop sizes only.

Initially, the text of each link is not visible. It must remain accessible, however, so you cannot use **display:none** here.

When the user mouses over a link, the text in **all menu items on that side** appears: the rest of the content slides over, with a transition.

The individual hovered-over menu item changes color to **firebrick.**

I gave you two hints earlier in this exercise. They will be very useful here.

Button (10 marks)

This also applies only to the tablet and desktop sizes.

A button appears in tablet and desktop views. When it is clicked, get both menus to display their text. When it is clicked again, set the layout back to not showing the text (the content will reflow).