Short Answer:

Answer the following questions with complete sentences in **your own words**. You are encouraged to conduct your own research online or through other methods before answering the questions. If you research online, please consult multiple sources before you write down your answers.

1. What is flow?
2. How are inline elements and block elements displayed in flow? Out of flow?
3. When is *float* useful?
4. What would happen if I set *float:left* to a block element?
5. What does *clear* do?
6. How is position useful?
7. What is the difference between *position:fixed* and *position:absolute*?
8. What does *z-index* do?
9. What is Flexbox?
10. What is a flex container? Can you explain some flex container properties?
11. What are flex items? Can you explain some flex item properties?
12. What is Grid?
13. What is a grid container? Can you explain some grid container properties?
14. What are grid items? Can you explain some grid item properties?
15. What are the major differences between CSS transitions and animations?
16. What is responsive web design (RWD)? What are some examples of RWD on a website? How do we achieve this?
17. What is a CSS preprocessor? What are the advantages and disadvantages, if any, to using them over plain CSS?
18. How do we declare a SCSS variable? How do we declare CSS variables?
19. What is the usage of '&' in SCSS?
20. What is @mixin? What are some use cases?
21. What is @extend? What are some use cases?
22. What is the usage of !important? What are some use cases?
23. Could you explain accessibility, usability, and inclusion? Give some examples of each one in terms of web design.
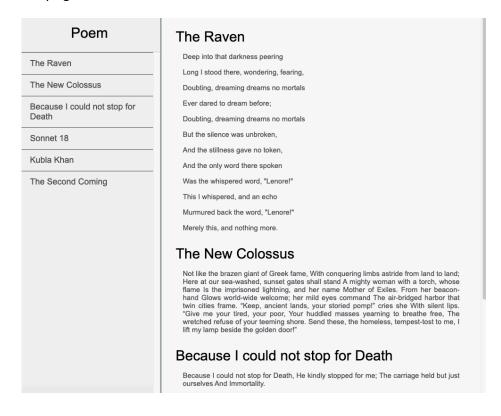

Coding Questions:

Use HTML and CSS to solve the following problems. You are highly encouraged to present more than one way to answer the questions. Please follow best practices when you write the code so that it would be easily readable, maintainable, and efficient. Clearly state your assumptions if you have any. You may discuss with others on the questions, but please write your own code.

1. Implement the following page using CSS Flexbox.

2. Implement the page from #1 using CSS Grid.

3. Implement the page from #1 using Bootstrap.

4. Design the following poem page using responsive web design.
In the right window, the poem contents should be displayed in the same order as the poem titles on the left navigation tab. When the user clicks on a poem title, e.g. 'The New Colossus', then the viewport should jump to that poem, making it appear at the top.

The poems in this example can be found at https://earlybirdbooks.com/most-famous-poems, but you can use any text that you'd like. Try to include enough poems or long-enough content so that the page needs a scrollbar.

When the window size is at a mobile device size, the layout should look like the image below. You only need breakpoints to account for *one* device type.