Assignment Questions 2

Q.1 What's Box Model in CSS ?
Ans: Imagine each element on a web page as a box. This box has different parts:

- Content: It's the actual content, like text or images, that goes inside the box.
- Padding: It's the space between the content and the edge of the box. It creates some breathing room around the content.
- Border: It's a line that surrounds the box. It separates the content from the padding and gives the box a defined shape.
- Margin: It's the space outside the box. It creates the gap between the box and other elements on the page.

So, the box model consists of these layers: content, padding, border, and margin. By adjusting these layers using CSS properties, we can control the size, spacing, and appearance of elements on a webpage.

Q.2 What are the Different Types of Selectors in CSS & what are the advantages of them?

Ans:1. **Element Selectors,** Advantages: Simple and target multiple elements with the same tag.

2. **Class Selectors**, Advantages: Target specific groups of elements and apply consistent styles.

3. **ID Selecto**r, Advantages: Target and style a specific unique element.

4. **Attribute Selectors**, Advantages: Flexibility to target elements based on various attributes and attribute values.

Q3. What is VW/VH ?
Ans: 1.**VW (Viewport Width)**: The VW unit represents a percentage of the viewport's width. For example, if you set an element's width to 50vw, it will occupy 50% of the viewport's width. So, regardless of the size of the viewport, the element will adjust its width accordingly.
2.VH (Viewport Height): The VH unit represents a percentage of the viewport's height. For instance, if you set an element's height to 50vh, it will occupy 50% of the viewport's height. Similar to VW, the element will adapt its height to the size of the viewport

Q.4 Whats difference between Inline, Inline Block and block ? **(3 Marks)**

Ans: Inline elements behave like text, they flow within the content and don't start on a new line. They can't have a specific width or height, and their spacing is mainly controlled horizontally.

Inline-block elements are similar to inline elements, but they can have a width, height, and vertical spacing. They still flow within the content, but they can be treated more like block elements in terms of size and spacing.

Block elements start on a new line and take up the full available width. They can have a specific width, height, and vertical spacing. They create distinct blocks on the page, like paragraphs or headings.

Q.5 How is Border-box different from Content Box? **(2 Marks)**
Ans:Border-box includes the element's padding and border within its specified width and height, while content-box does not and adds them on top of the specified dimensions.

Q.6 What's z-index and How does it Function ? **(2 Marks)**

Ans :Z-index determines which elements appear in front of or behind other elements on a webpage. It works like a layering system, where elements with higher values are placed on top of elements with lower values. This allows you to control the order in which elements are displayed and create visual depth in your webpage**.**

Q.6 What's Grid & Flex and difference between them? **(5 Marks)**
Ans :The main difference between Grid and Flexbox lies in their layout capabilities.
Grid is a two-dimensional layout system that allows for complex grid-based layouts with rows and columns. It offers precise control over item placement and alignment in both dimensions.

Flexbox is a one-dimensional layout system that focuses on arranging items in a single row or column. It simplifies the distribution of space, alignment, and order of elements within the flex container. Flexbox is best suited for simpler, one-dimensional layouts.

Q.7 Difference between absolute and relative and sticky and fixed position explain with example. **(5 Marks)**
Ans:Absolute Positioning vs. Relative Positioning:

.**Absolute positioning**: The element is taken out of the normal flow and positioned based on its closest positioned ancestor or the window. It can be positioned precisely using top, bottom, left, and right properties.

Example: Placing a popup box exactly 20 pixels from the top and 20 pixels from the left of its parent container.

```
<div style="position: relative;">
  <p style="position: absolute; top: 20px; left: 20px;">Absolute positioned paragraph</p>
</div>
```

Relative positioning: The element remains in the normal flow but can be offset from its original position. It is positioned relative to its normal position using top, bottom, left, and right properties.

Example: Moving a paragraph 10 pixels down and 10 pixels to the right from its original position.

```
<div style="position: relative;">
  <p style="position: relative; top: 10px; left: 10px;">Relative positioned paragraph</p>
</div>
```

2. Sticky Positioning vs. Fixed Positioning:

Sticky positioning: The element behaves normally in the flow until a specific scroll position is reached. Once that point is reached during scrolling, the element becomes "stuck" and remains in a fixed position within its container.

Example: Having a navigation bar that remains at the top of the screen when scrolling down until a certain point, and then sticks to that position.

```
<div style="height: 200px; overflow-y: scroll;">
  <p style="position: sticky; top: 20px;">Sticky positioned paragraph</p>
</div>
```

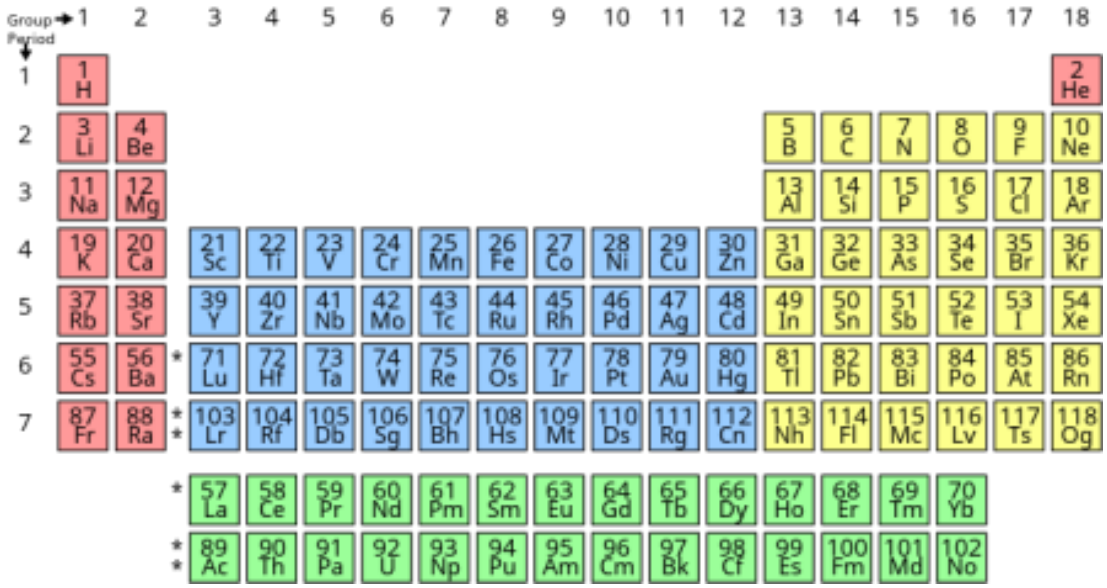
Fixed positioning: The element is positioned relative to the viewport (window) and remains fixed in that position even when scrolling.

Example: Having a back-to-top button that always stays in the bottom-right corner of the screen, regardless of scrolling.

```
<p style="position: fixed; top: 20px; left: 20px;">Fixed positioned paragraph</p>
```

In simple terms, absolute and relative positioning allow you to move elements around, while sticky and fixed positioning help elements stay in specific positions as you scroll or interact with the page.

Q.8 Build Periodic Table as shown in the below image **(10 Marks)**



Ans: link:
https://github.com/Devpanjum/FSJS/tree/main/Interview/CSS/Q8%20Periodic%20Table

Q.10 Build Responsive Layout both desktop and mobile and Tablet, see below image for reference ?
Ans
https://github.com/Devpanjum/FSJS/tree/main/DSA/Mock%20test/Web%20devlopment/Assignment%202/Q10