Day 2 → Exploring Advanced Layout Techniques: CSS Flexbox

1. Introduction to CSS Flexbox:

CSS Flexbox is a layout model that provides a flexible way to distribute space among elements within a container. It simplifies the process of creating one-dimensional layouts, such as horizontal or vertical alignment, and handles complex alignment scenarios.

Example:

```css
.container {
  display: flex;
  justify-content: center;
  align-items: center;
}
```

2. Exploring Flexbox Properties:

Flexbox offers various properties that control the behavior and appearance of flex items. We will cover key properties such as flex-direction, flex-wrap, justify-content, align-items, and align-self. Understanding these properties enables precise control over the arrangement and positioning of elements.

Example:

```css
.container {
```

```
    display: flex;
    flex-direction: row;
    justify-content: space-between;
    align-items: center;
  }
  .item {
    flex: 1;
  }
  ```
```

Flex Container Properties
- Present key properties for the flex container:
  - `display: flex` to enable Flexbox layout.
  - `flex-direction` to set the direction of the main axis (row or column).
  - `justify-content` to align items along the main axis.
  - `align-items` to align items along the cross axis.

Flex Item Properties
- Present important properties for flex items:
  - `flex-grow` to define how much the item can grow to fill available space.
  - `flex-shrink` to determine how much the item can shrink when space is limited.
  - `flex-basis` to specify the initial size of the item.
  - `flex` (shorthand) to combine `flex-grow`, `flex-shrink`, and `flex-basis`.
```

Flexbox Alignment
- Explain alignment within the flex container:
** - `justify-content` for horizontal alignment along the main axis.**
** - `align-items` for vertical alignment along the cross axis.**
** - `align-self` to override alignment for individual items.**

Flexbox Layout Examples
- Showcase practical examples of common Flexbox layouts, such as:
** - Horizontally centered content.**
** - Vertically aligned content.**
** - Equal-width columns.**
** - Responsive grid layouts.**

Responsive Design with Flexbox
- Discuss how Flexbox enables responsive design:
** - Explain how flex items can automatically wrap or collapse based on available space.**

Code:
https://github.com/Vijayanandr2000/zybisys_training/tree/main/CSS