index.html

```html
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>CSS Layout & Positioning Techniques</title>
</head>

<body>
    <h1>CSS Layout & Positioning Techniques</h1>

    <p>CSS layout refers to the process of positioning and arranging elements on a web page using Cascading Style Sheets (CSS). The layout of a web page is an essential aspect of its design, as it affects the visual hierarchy of the content and its usability.</p>

    <p>CSS Positioning refers to the process of controlling the position of elements on a webpage using CSS. Positioning is a fundamental aspect of web design, as it allows designers to create custom layouts and control the visual hierarchy of content. </p>

    <h2>Examples of CSS layout techniques</h2>

    <ul>
        <li>Normal Layout: The normal layout is the default layout that web browsers use to display web pages. In this layout, elements are placed in the order in which they appear in the HTML document, and their positions are determined by their default positioning properties.</li>

        <li>Float: The float layout technique involves positioning elements by "floating" them to one side of their container. This is typically used to position images or other elements next to text or other content.</li>

        <li>Flexbox: The flex layout technique is based on a flexible box model, which allows elements to be arranged in rows or columns. This technique is ideal for creating responsive layouts that can adapt to different screen sizes and device orientations. Flexbox offers a lot of control over the position and sizing of elements, making it a popular choice for modern web design.</li>

```

```
        <li>Grid: The grid layout technique is similar to the flex layout, but with more advanced features for creating complex, multi-dimensional layouts. With grid layout, elements are positioned on a grid of rows and columns, and can span multiple rows or columns. This allows for precise control over the layout of content, making it a powerful tool for web designers. Grid layout is also ideal for creating responsive layouts, as elements can be repositioned and resized automatically based on the screen size.</li>
    </ul>

    <h2>Examples of CSS Positioning techniques</h2>

    <ul>
        <li>Static: The static positioning technique places an element relative to the normal flow of the document. This is the default positioning method for web pages.</li>

        <li>Relative: The relative positioning technique places an element relative to its normal position. This positioning method is useful when you want to position an element relative to its current position.</li>

        <li>Absolute: The absolute positioning technique places an element absolutely relative to the viewport. This positioning method is useful when you want to position an element relative to the viewport.</li>

        <li>Fixed: The fixed positioning technique places an element relative to the viewport. This positioning method is useful when you want to position an element relative to the viewport.</li>

        <li>Sticky: Sticky positioning is a hybrid of relative and fixed positioning. This technique allows an element to behave as if it is positioned statically until it reaches a certain threshold, after which it becomes fixed. This technique is often used for creating sticky headers, sidebars, or navigation menus.</li>

    </ul>

</body>

</html>
```

style.css

```css
:root {
    --heading-color: rgb(95, 11, 192);
    --list-color: rgb(223, 17, 17);
}

* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: 'Times New Roman', Times, serif;
}

h1, h2 {
    text-align: center;
    margin-top: 0.7rem;
    color: var(--heading-color);
}

p, li {
    margin-top: 0.5rem;
    font-size: 1rem;
    font-weight: bold;
}

ul li {
    color: var(--list-color);
}
```

CSS Layout & Positioning Techniques

CSS layout refers to the process of positioning and arranging elements on a web page using Cascading Style Sheets (CSS). The layout of a web page is an essential aspect of its design, as it affects the visual hierarchy of the content and its usability.

CSS Positioning refers to the process of controlling the position of elements on a webpage using CSS. Positioning is a fundamental aspect of web design, as it allows designers to create custom layouts and control the visual hierarchy of content.

Examples of CSS layout techniques

Normal Layout: The normal layout is the default layout that web browsers use to display web pages. In this layout, elements are placed in the order in which they appear in the HTML document, and their positions are determined by their default positioning properties.

Float: The float layout technique involves positioning elements by "floating" them to one side of their container. This is typically used to position images or other elements next to text or other content.

Flexbox: The flex layout technique is based on a flexible box model, which allows elements to be arranged in rows or columns. This technique is ideal for creating responsive layouts that can adapt to different screen sizes and device orientations. Flexbox offers a lot of control over the position and sizing of elements, making it a popular choice for modern web design.

Grid: The grid layout technique is similar to the flex layout, but with more advanced features for creating complex, multi-dimensional layouts. With grid layout, elements are positioned on a grid of rows and columns, and can span multiple rows or columns. This allows for precise control over the layout of content, making it a powerful tool for web designers. Grid layout is also ideal for creating responsive layouts, as elements can be repositioned and resized automatically based on the screen size.

Examples of CSS Positioning techniques

Static: The static positioning technique places an element relative to the normal flow of the document. This is the default positioning method for web pages.

Relative: The relative positioning technique places an element relative to its normal position. This positioning method is useful when you want to position an element relative to its current position.

Absolute: The absolute positioning technique places an element absolutely relative to the viewport. This positioning method is useful when you want to position an element relative to the viewport.

Fixed: The fixed positioning technique places an element relative to the viewport. This positioning method is useful when you want to position an element relative to the viewport.

Sticky: Sticky positioning is a hybrid of relative and fixed positioning. This technique allows an element to behave as if it is positioned statically until it reaches a certain threshold, after which it becomes fixed. This technique is often used for creating sticky headers, sidebars, or navigation menus.