**Experiment 6 Documentation**

**Experiment Title: Complex Layout with CSS**

**1. Problem Statement**

**Create a responsive webpage layout using HTML and CSS that includes a header, footer, sidebars, and a main content area. The purpose is to understand how to utilize CSS Grid for layout design.**

**2. Steps and Explanation**

**1. Setting Up the HTML Structure**

* **Begin with the basic HTML tags: <!DOCTYPE html>, <html>, <head>, and <body>.**
* **Include meta tags for character set and viewport settings in the <head> section.**
* **Use a <title> tag to set the page's title to "Complex Layout with CSS."**

**2. Adding CSS Styles**

* **Create a <style> section within the <head> to define the CSS rules.**
* **Set the body’s font and background color for better aesthetics.**

**3. Defining the Grid Layout**

* **Use a CSS grid to create the layout with display: grid.**
* **Define grid-template-areas to specify the layout of the header, footer, sidebars, and content area.**
* **Set grid-template-columns for defining the width of the columns.**

**4. Styling Header and Footer**

* **Use specific classes for the header and footer to style their background color, text alignment, padding, and box shadows.**

**5. Adding Sidebar and Content Area Styles**

* **Create styles for the left sidebar, main content area, and right sidebar, including background colors, padding, and shadows for depth.**

**6. Implementing Responsive Design**

* **Use media queries to adjust the layout for smaller screens, ensuring that elements stack vertically instead of side by side.**

**3. Expected Output**

* **The webpage displays a structured layout with a header, footer, left sidebar, right sidebar, and main content area.**
* **On larger screens, the layout will be horizontal, and on smaller screens (less than 768px), it will adjust to a vertical layout.**

