Here’s an \*\*in-depth problem statement\*\* for the CSS Grid Layout assignment, styled like a coding challenge from \*\*LeetCode\*\* or \*\*Codecademy\*\*:

---

### \*\*Problem: Two-Column Webpage Layout with CSS Grid\*\*

\*\*Problem Statement\*\*:

You are tasked with designing a simple two-column webpage layout using CSS Grid. The layout should consist of a sidebar and a main content area, with appropriate styling to enhance the visual presentation.

\*\*Requirements\*\*:

1. The webpage should have a header, a sidebar, a main content area, and a footer.

2. Use CSS Grid to create a layout with two columns:

- The sidebar should take up \*\*25%\*\* of the width.

- The main content area should take up \*\*75%\*\* of the width.

3. The layout should include:

- A header at the top that spans both columns.

- A footer at the bottom that also spans both columns.

4. The sidebar should contain links or menu items (e.g., Home, About, Services).

5. The main content area should include a sample article or paragraph.

6. Ensure that the layout is responsive and adjusts gracefully for smaller screen sizes.

\*\*Constraints\*\*:

- You may not use frameworks like Bootstrap; use pure HTML and CSS.

- Use semantic HTML elements where appropriate (e.g., `<header>`, `<nav>`, `<main>`, `<footer>`).

\*\*Example\*\*:

```html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Two-Column Layout</title>

<style>

body {

margin: 0;

font-family: Arial, sans-serif;

}

.container {

display: grid;

grid-template-columns: 1fr 3fr; /\* Sidebar 25%, Main content 75% \*/

grid-template-rows: auto 1fr auto; /\* Header, Content, Footer \*/

height: 100vh; /\* Full viewport height \*/

}

header {

grid-column: 1 / -1; /\* Header spans both columns \*/

background-color: #4CAF50;

color: white;

text-align: center;

padding: 1rem;

}

.sidebar {

background-color: #f4f4f4;

padding: 1rem;

border-right: 1px solid #ccc;

}

.main {

padding: 1rem;

}

footer {

grid-column: 1 / -1; /\* Footer spans both columns \*/

background-color: #4CAF50;

color: white;

text-align: center;

padding: 1rem;

position: relative;

bottom: 0;

width: 100%;

}

/\* Media Query for small screens \*/

@media (max-width: 768px) {

.container {

grid-template-columns: 1fr; /\* Single column layout on small screens \*/

}

}

</style>

</head>

<body>

<div class="container">

<header>

<h1>My Website Header</h1>

</header>

<nav class="sidebar">

<h2>Sidebar Menu</h2>

<ul>

<li><a href="#home">Home</a></li>

<li><a href="#about">About</a></li>

<li><a href="#services">Services</a></li>

</ul>

</nav>

<main class="main">

<h2>Main Content Area</h2>

<p>This is where the main content goes. You can write articles, blogs, or other content here.</p>

</main>

<footer>

<p>Footer Information</p>

</footer>

</div>

</body>

</html>

```

\*\*Expected Output\*\*:

- A two-column layout with a sidebar on the left and a main content area on the right. The header and footer should span the entire width of the page. The layout should adjust to a single-column format on smaller screens, maintaining readability and accessibility.

---

This problem statement clearly outlines the task and provides requirements and examples, guiding learners through creating a responsive two-column webpage layout using CSS Grid.