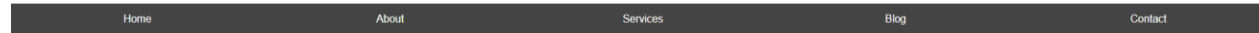


Creating a **CSS Grid Layout for a Menu** is a great way to organize navigation items, and CSS Grid provides flexibility for both horizontal and vertical layouts. Below, I'll show how to create a responsive horizontal menu and a vertical sidebar-style menu using CSS Grid.

## 1. Horizontal Menu Layout with CSS Grid



A simple, horizontal navigation bar that is responsive and adjusts on smaller screens.

### HTML Structure

html code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Grid Horizontal Menu</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <nav class="navbar">
      <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Services</a></li>
        <li><a href="#">Blog</a></li>
        <li><a href="#">Contact</a></li>
      </ul>
    </nav>
  </header>
</body>
</html>
```

### CSS (styles.css)

Css code

```
/* Basic reset */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

body {
  font-family: Arial, sans-serif;
}

.navbar {
  display: grid;
  grid-template-columns: repeat(5, 1fr); /* Five equally spaced columns */
  background-color: #333;
}
```

```

.navbar ul {
  display: contents; /* Removes the ul wrapper's grid container role */
}

.navbar li {
  list-style: none;
  text-align: center;
}

.navbar a {
  display: block;
  padding: 14px 20px;
  text-decoration: none;
  color: white;
  background-color: #444;
  transition: background-color 0.3s;
}

.navbar a:hover {
  background-color: #4CAF50; /* Change color on hover */
}

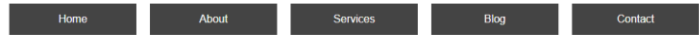
@media (max-width: 768px) {
  .navbar {
    grid-template-columns: 1fr 1fr; /* Stack in two columns on small screens
*/
  }
}

```

## Explanation:

- The `.navbar` is a grid container with `grid-template-columns` set to `repeat(5, 1fr)`, which means five equal-width columns for the menu items.
- Each `<li>` represents a grid item, and the links inside `<a>` elements fill each column.
- The `@media` query adjusts the layout for smaller screens (`max-width: 768px`), stacking the menu items in two columns.

Css Grid layout for header :-



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Grid Horizontal Menu</title>
  <style>
    /* Basic reset */
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
    }

    body {
      font-family: Arial, sans-serif;
    }

    .header
```

```
{
display: grid;
grid-template-columns: 1fr 2fr;
padding:20px;
}

.image {
padding: 10px;

}

.navbar {

padding :20px;
}

ul {
display: grid;
grid-template-columns: repeat(5, 1fr); /* Five equally spaced columns */

gap:20px;
}

ul li {
list-style: none;
text-align: center;
}
```

```
ul li a {  
  display: block;  
  padding: 14px 20px;  
  text-decoration: none;  
  color: white;  
  background-color: #444;  
  transition: background-color 0.3s;  
}
```

```
.navbar a:hover {  
  background-color: #4CAF50; /* Change color on hover */  
}
```

```
@media (max-width: 768px) {  
  .navbar {  
    grid-template-columns: 1fr 1fr; /* Stack in two columns on small screens */  
  }  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div class="header">
```

```

```

```
<div class="navbar">
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About</a></li>
    <li><a href="#">Services</a></li>
    <li><a href="#">Blog</a></li>
    <li><a href="#">Contact</a></li>
  </ul>
</div>
</div>
</body>
</html>
```