

Media Query

Assignment Solution



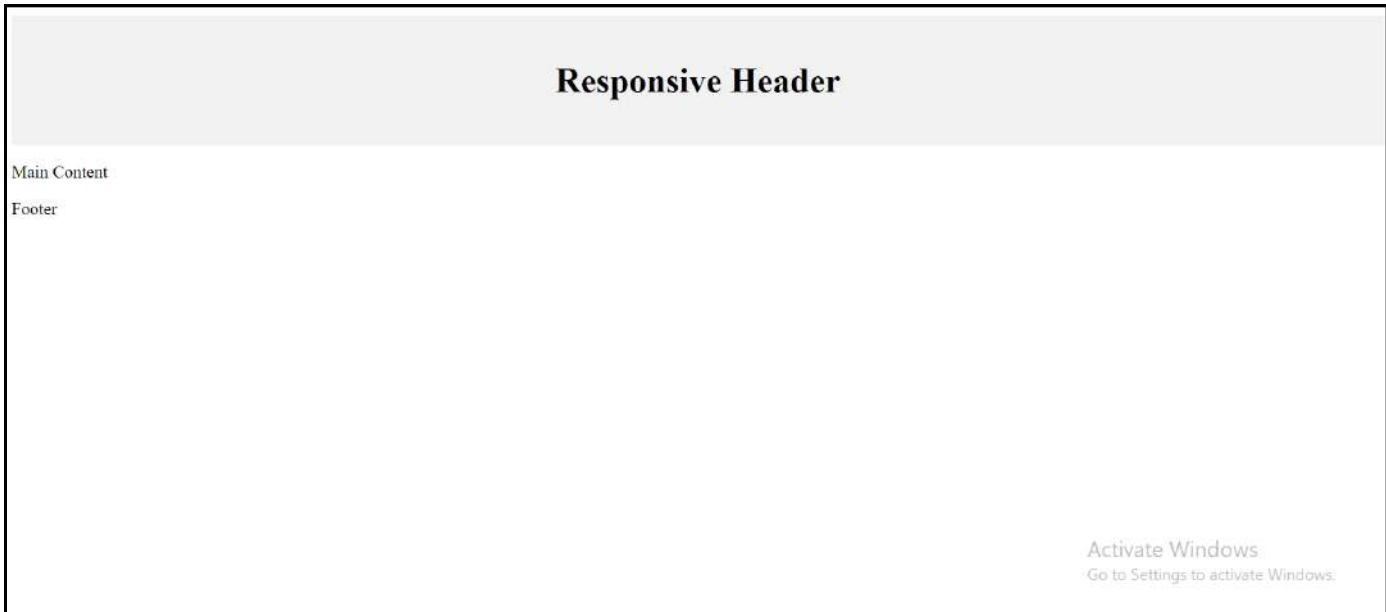
Task 1:

Problem Statement

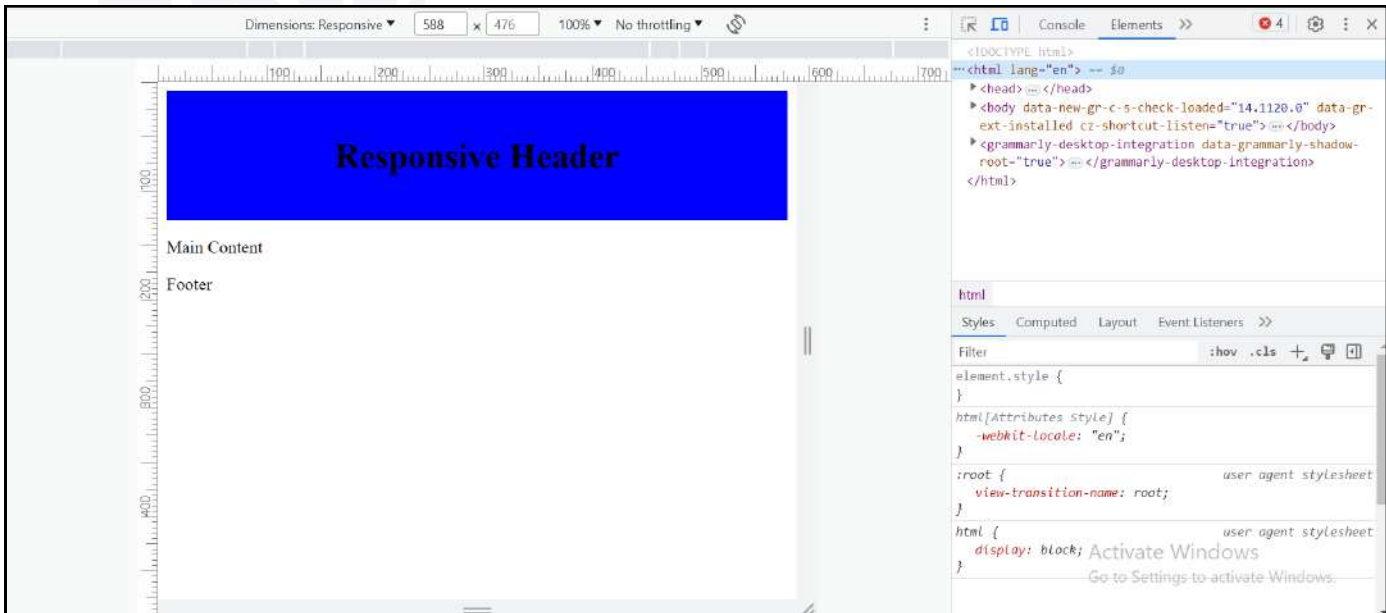
Create a simple webpage with a header, a main content section, and a footer. Apply a media query that changes the background color of the header to blue when the screen width is less than 600px.

Expected Outputs

Normal Output



Responsive Output



Solution

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <style>
      header {
        background-color: #f2f2f2;
        padding: 20px;
        text-align: center;
      }
      @media screen and (max-width: 600px) {
        header {
          background-color: blue;
        }
      }
    </style>
  </head>
  <body>
    <header>
      <h1>Responsive Header</h1>
    </header>
    <main>
      <p>Main Content</p>
    </main>
    <footer>
      <p>Footer</p>
    </footer>
  </body>
</html>
```

Task 2:

Problem Statement

Create an image gallery with three images in a row. Use media queries to adjust the layout to two images in a row for screens smaller than 800px and one image in a row for screens smaller than 500px.

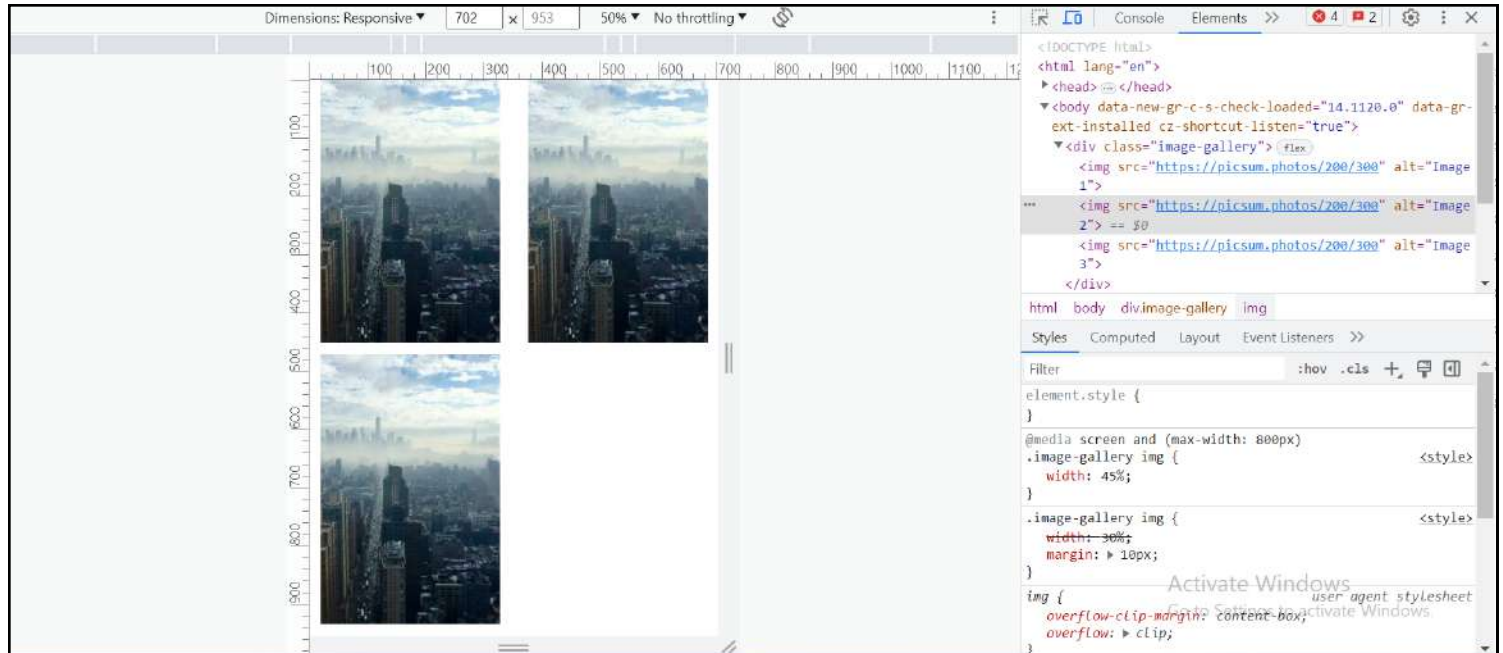
For image use this link ([Image Link](#))

Expected Output

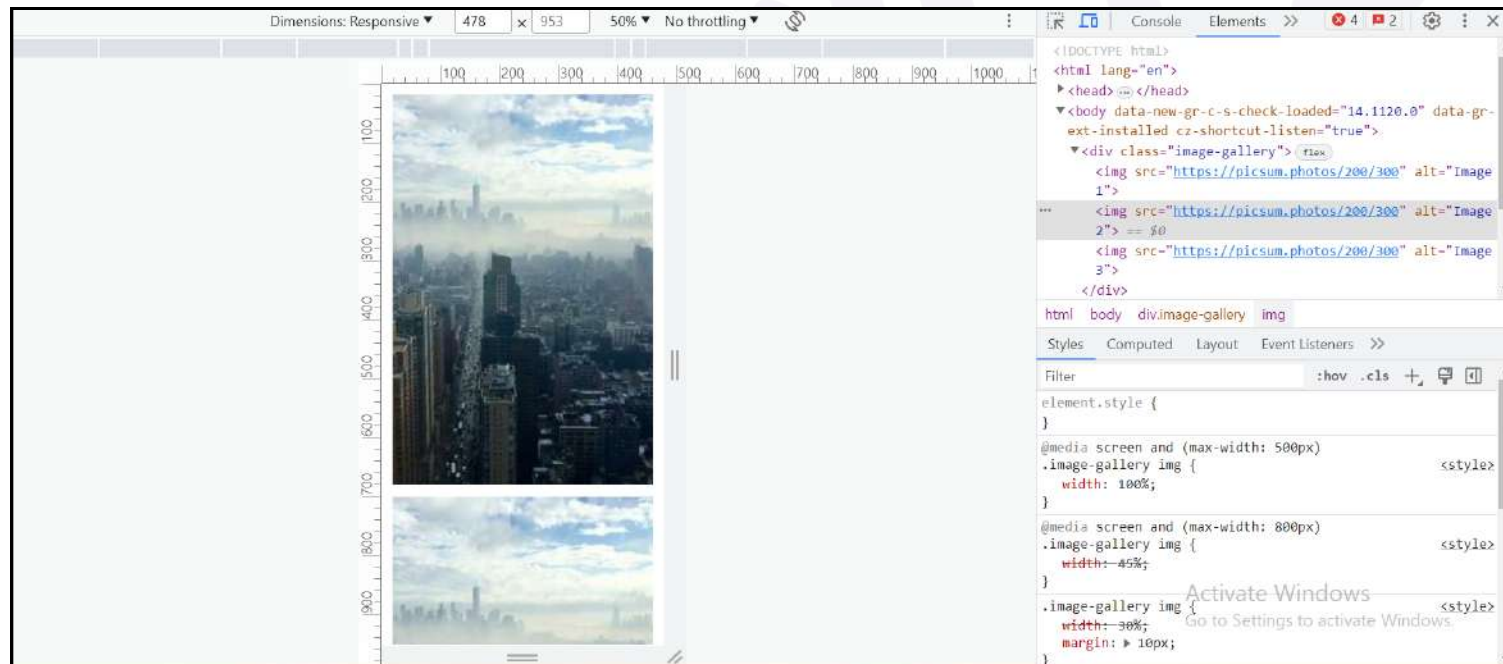
First Output



Second Output



Third Output



Solution

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <style>
    .image-gallery {
      display: flex;
      flex-wrap: wrap;
      justify-content: space-between;
    }
    .image-gallery img {
      width: 30%;
      margin: 10px;
    }
    @media screen and (max-width: 800px) {
      .image-gallery img {
        width: 45%;
      }
    }
    @media screen and (max-width: 500px) {
      .image-gallery img {
        width: 100%;
      }
    }
  </style>
  <body>
    <div class="image-gallery">
      
      
      
    </div>
  </body>
</html>
```

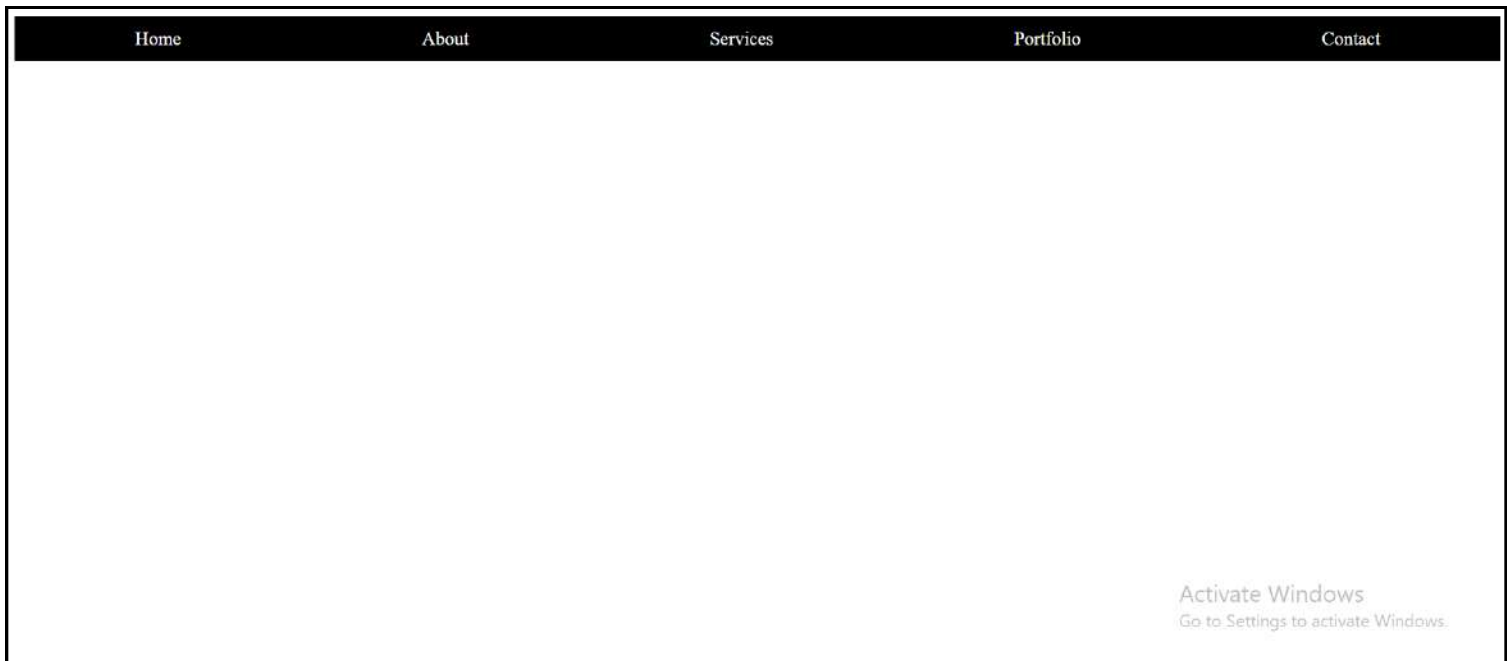
Task 3:

Problem Statement

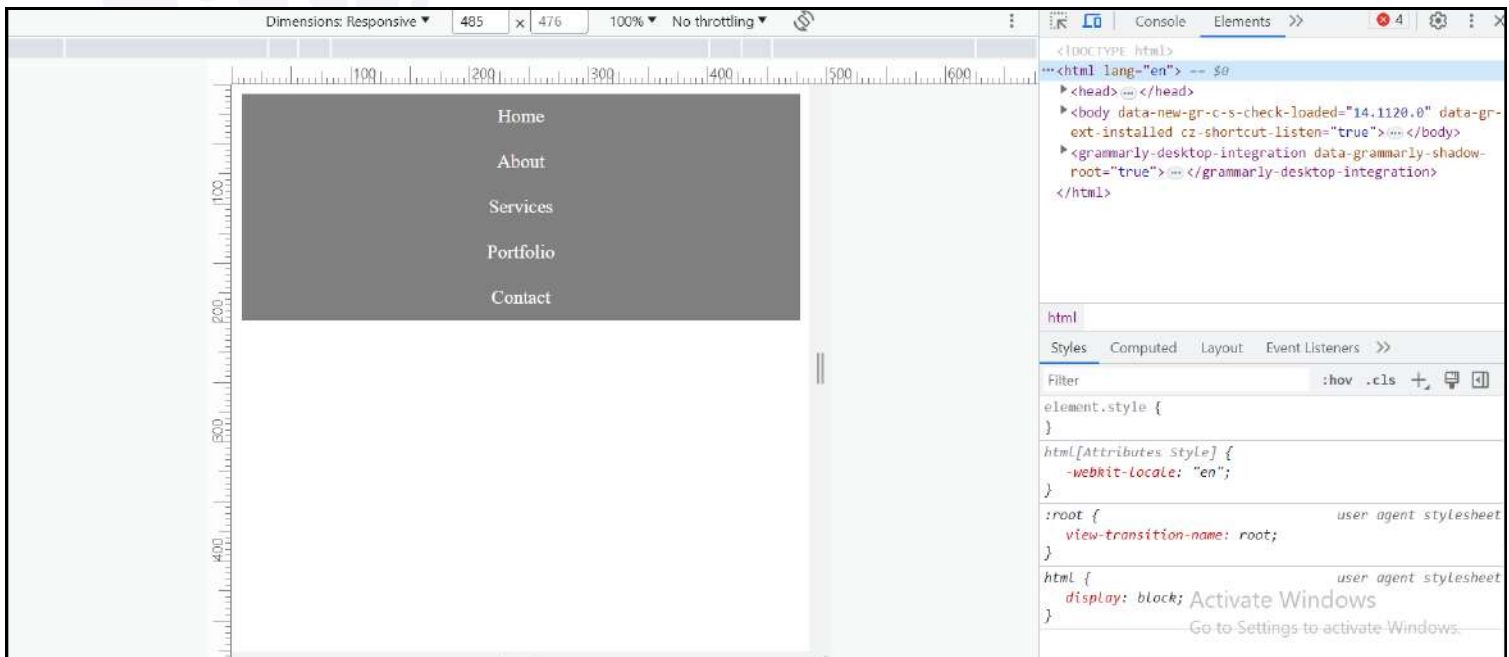
Create a navigation bar with five links. Apply a media query that converts the navigation into a vertical list for screens smaller than 768px and also changes the background color to gray.

Expected Outputs

Normal Output



Responsive Output



Solution

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <style>
    .navbar {
      display: flex;
      justify-content: space-around;
      background-color: black;
    }
    .navbar a {
      text-decoration: none;
      padding: 10px;
      color: white;
    }
    @media screen and (max-width: 768px) {
      .navbar {
        flex-direction: column;
        align-items: center;
        background-color: gray;
      }
    }
  </style>
  <body>
    <nav class="navbar">
      <a href="#">Home</a>
      <a href="#">About</a>
      <a href="#">Services</a>
      <a href="#">Portfolio</a>
      <a href="#">Contact</a>
    </nav>
  </body>
</html>
```

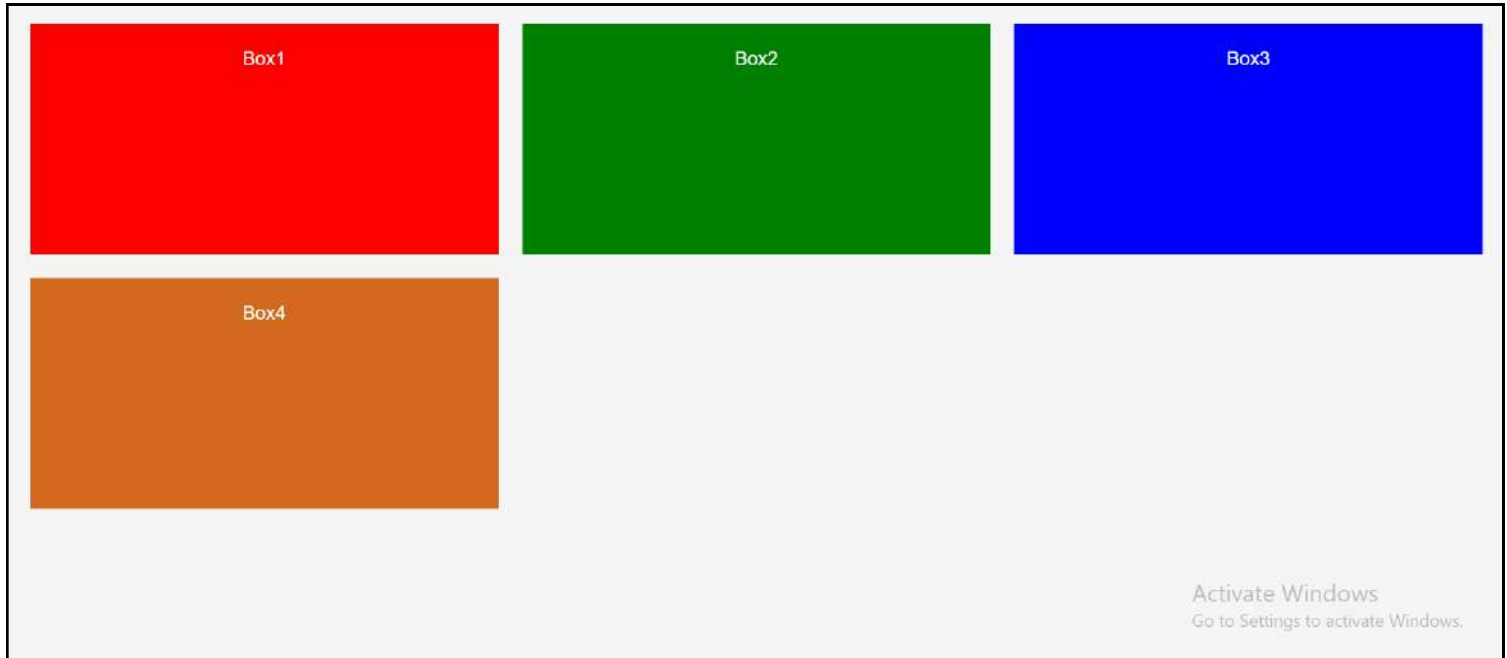

Task 4:

Problem Statement

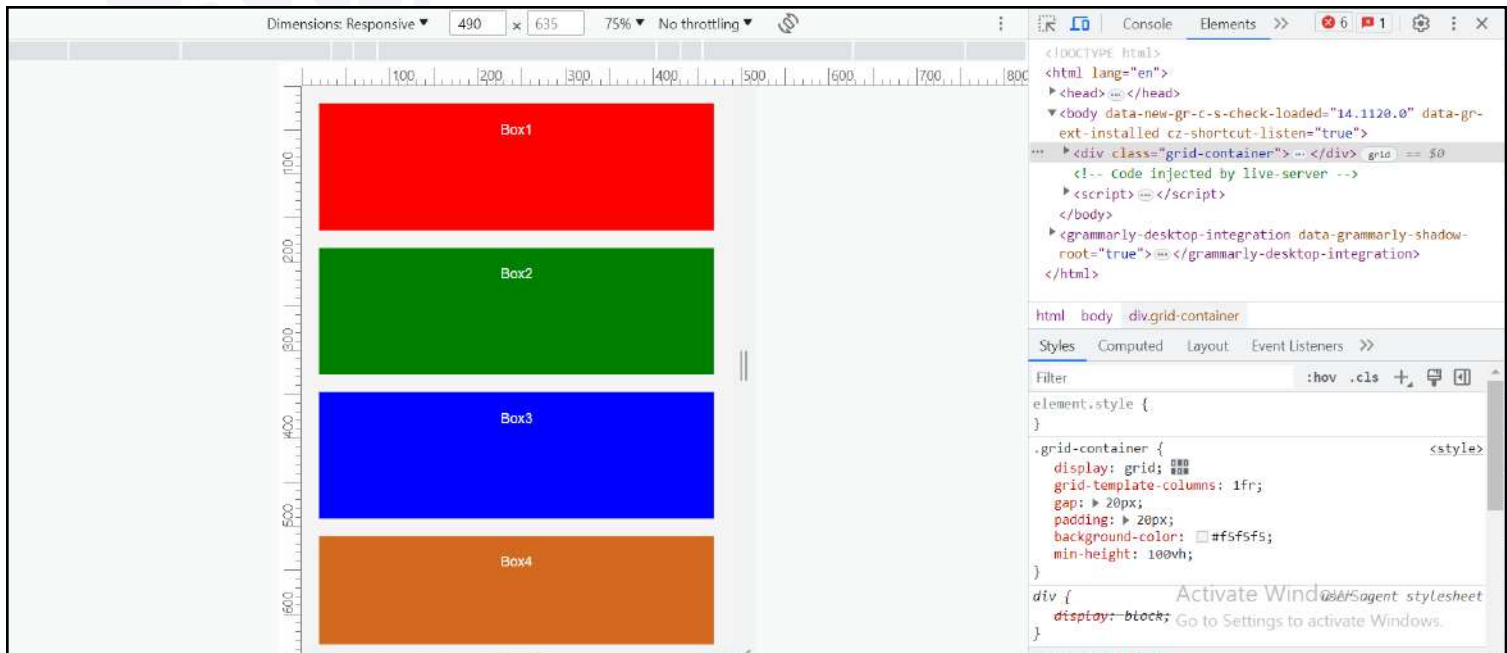
Create a grid based layout containing four grid items which arranges them as the screen size changes.

Expected Outputs

Normal Output



Responsive Output



Solution

```
<!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="styles.css" />
    <title>Responsive Grid Layout Example</title>
  </head>
  <style>
    body {
      margin: 0;
      font-family: Arial, sans-serif;
    }

    .grid-container {
      display: grid;
      grid-template-columns: 1fr;
      gap: 20px;
      padding: 20px;
      background-color: #f5f5f5;
      min-height: 100vh;
    }

    .item {
      background-color: #ddd;
      padding: 20px;
      text-align: center;
    }

    .header {
      background-color: red;
      color: white;
    }

    .sidebar {
      background-color: green;
      color: white;
    }

    .main {
      background-color: blue;
      color: white;
    }

    .footer {
      background-color: chocolate;
      color: white;
    }
  </style>
</html>
```

```
/* Media Queries */
@media screen and (min-width: 768px) {
  .grid-container {
    grid-template-columns: repeat(3, 1fr);
    grid-template-areas:
      "header header header"
      "sidebar main main"
      "footer footer footer";
  }
}
</style>
<body>
  <div class="grid-container">
    <div class="item header">Box1</div>
    <div class="item sidebar">Box2</div>
    <div class="item main">Box3</div>
    <div class="item footer">Box4</div>
  </div>
</body>
</html>
```



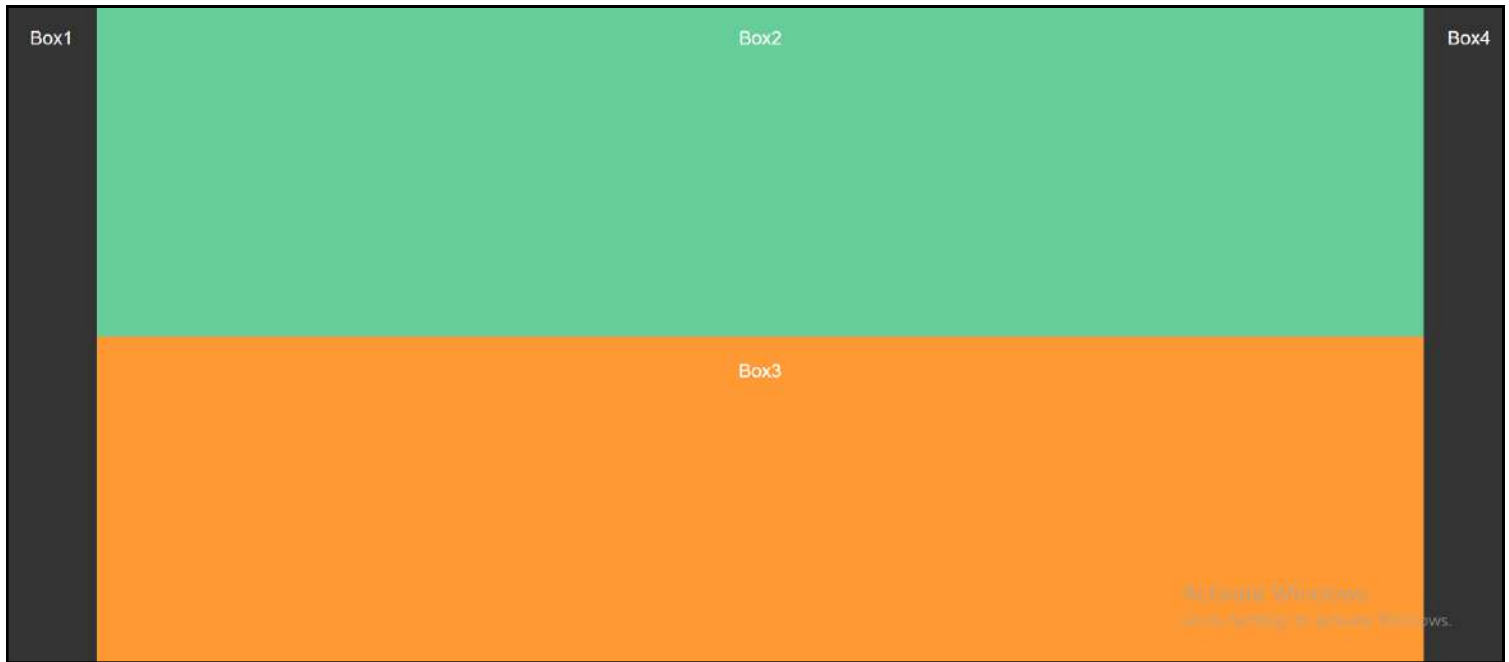
Task 5:

Problem Statement

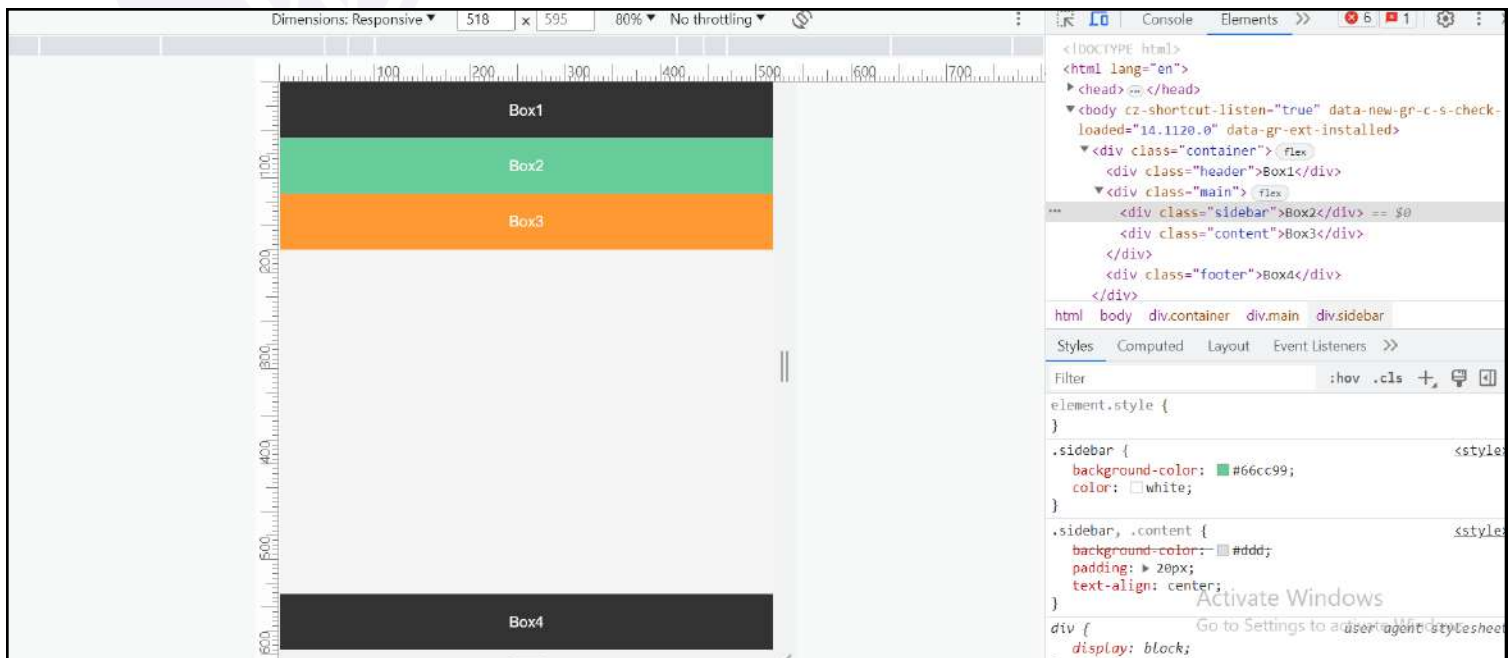
Create the below given layout using the flexbox in css, which should adapt itself on mobile screen as per given below output.

Expected Outputs

Normal Output



Responsive Output



Solution

```
<!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="styles.css" />
    <title>Responsive Flex Layout Example</title>
  </head>
  <style>
    body {
      margin: 0;
      font-family: Arial, sans-serif;
    }

    .container {
      display: flex;
      flex-direction: column;
      min-height: 100vh;
      background-color: #f5f5f5;
    }

    .header,
    .footer {
      background-color: #333;
      color: white;
      padding: 20px;
      text-align: center;
    }

    .main {
      display: flex;
      flex-direction: column;
      flex: 1;
    }

    .sidebar,
    .content {
      background-color: #ddd;
      padding: 20px;
      text-align: center;
    }

    .sidebar {
      background-color: #66cc99;
      color: white;
    }
  </style>
</html>
```

```
.content {
  background-color: #ff9933;
  color: white;
}

/* Media Queries */
@media screen and (min-width: 768px) {
  .container {
    flex-direction: row;
  }

  .sidebar,
  .content {
    flex: 1;
  }
}
</style>
<body>
  <div class="container">
    <div class="header">Header</div>
    <div class="main">
      <div class="sidebar">Sidebar</div>
      <div class="content">Main Content</div>
    </div>
    <div class="footer">Footer</div>
  </div>
</body>
</html>
```