

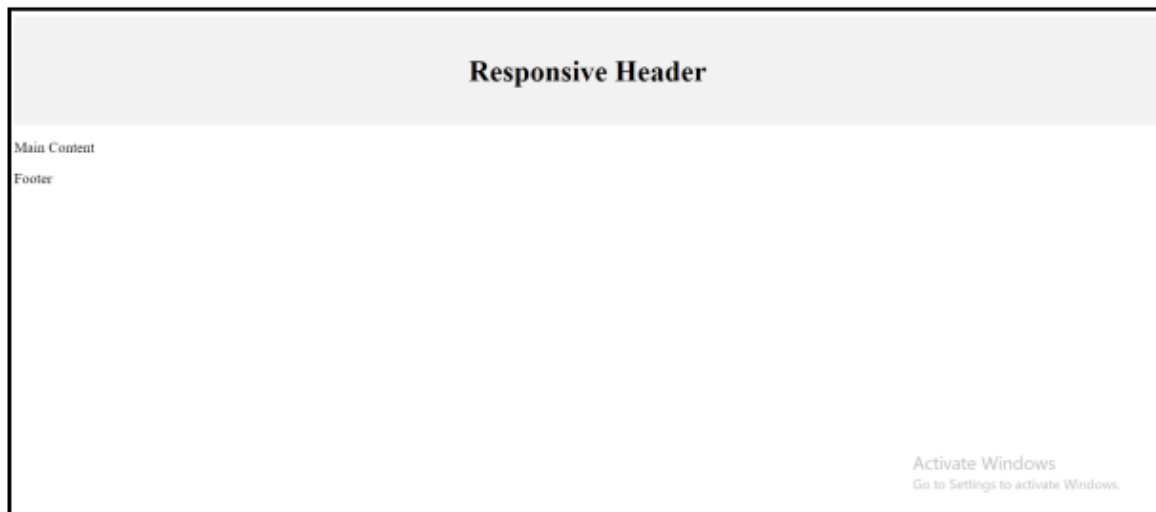
Task 1:

Problem Statement

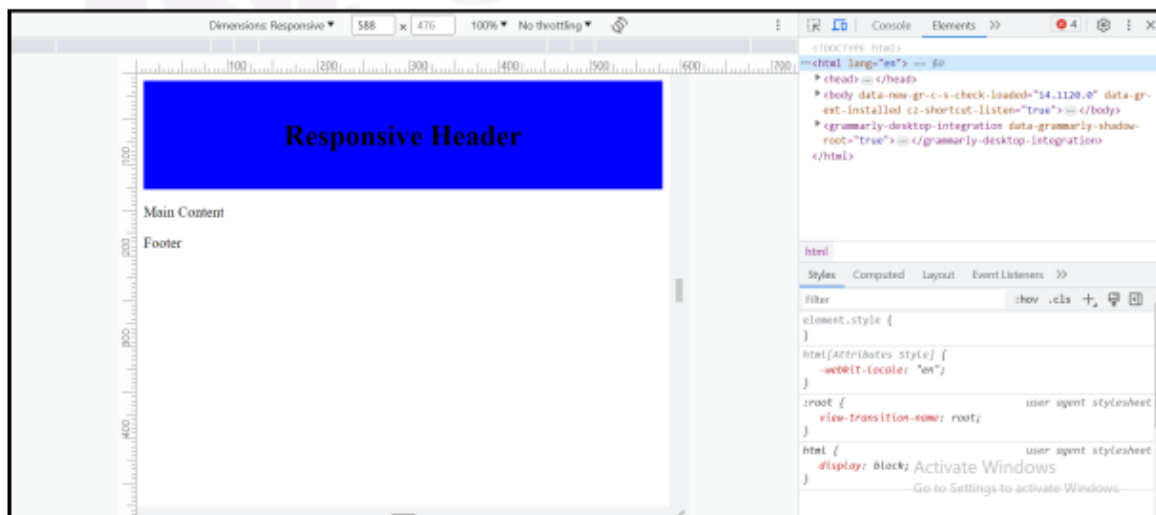
Create a simple webpage with a header, a main content section, and a footer. Apply a media query that changes the background color of the header to blue when the screen width is less than 600px.

Expected Outputs

Normal Output



Responsive Output



Ans: `<!DOCTYPE html>`

`<html lang="en">`

`<head>`

`<meta charset="UTF-8">`

```

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Responsive Webpage</title>
<style>
  body {
    font-family: sans-serif;
    margin: 0;
    padding: 0;
  }

  header {
    background-color: #f0f0f0;
    padding: 20px;
    text-align: center;
  }
  @media (max-width: 600px) {
    header {
      background-color: blue;
    }
  }
</style>
</head>
<body>
  <header>
    <h1>Responsive Header</h1>
  </header>

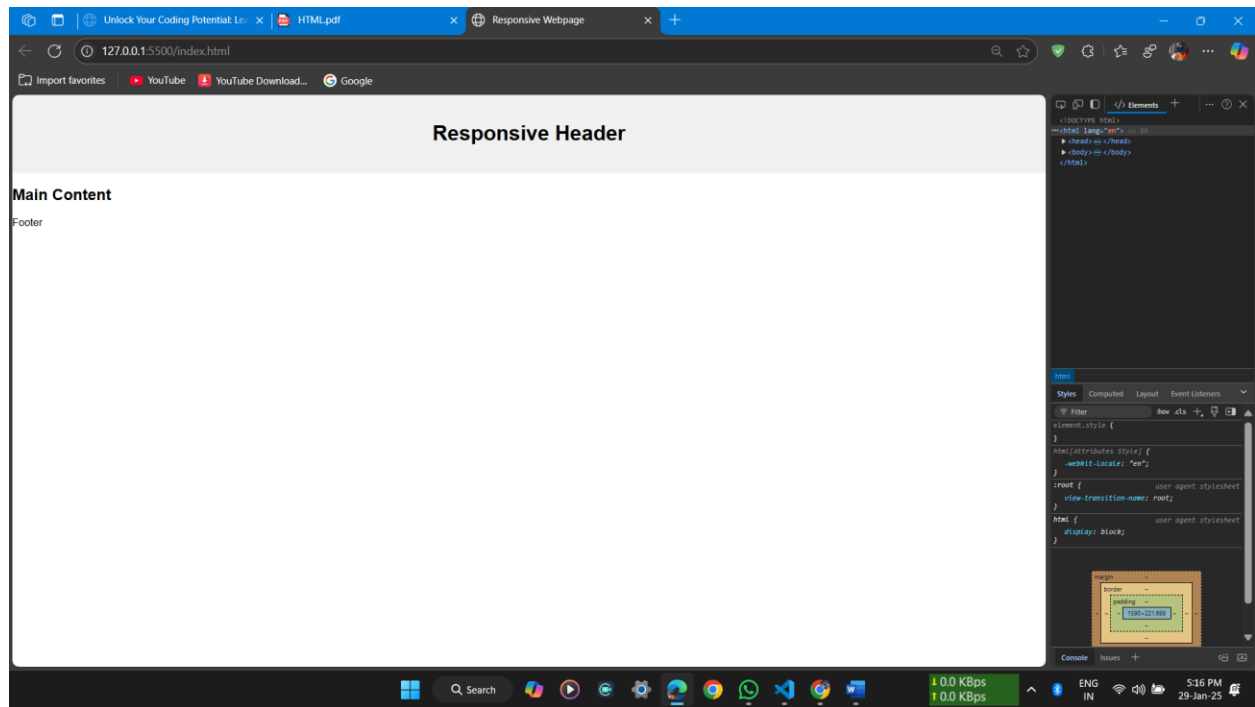
  <main>
    <section>
      <h2>Main Content</h2>

    </section>
  </main>

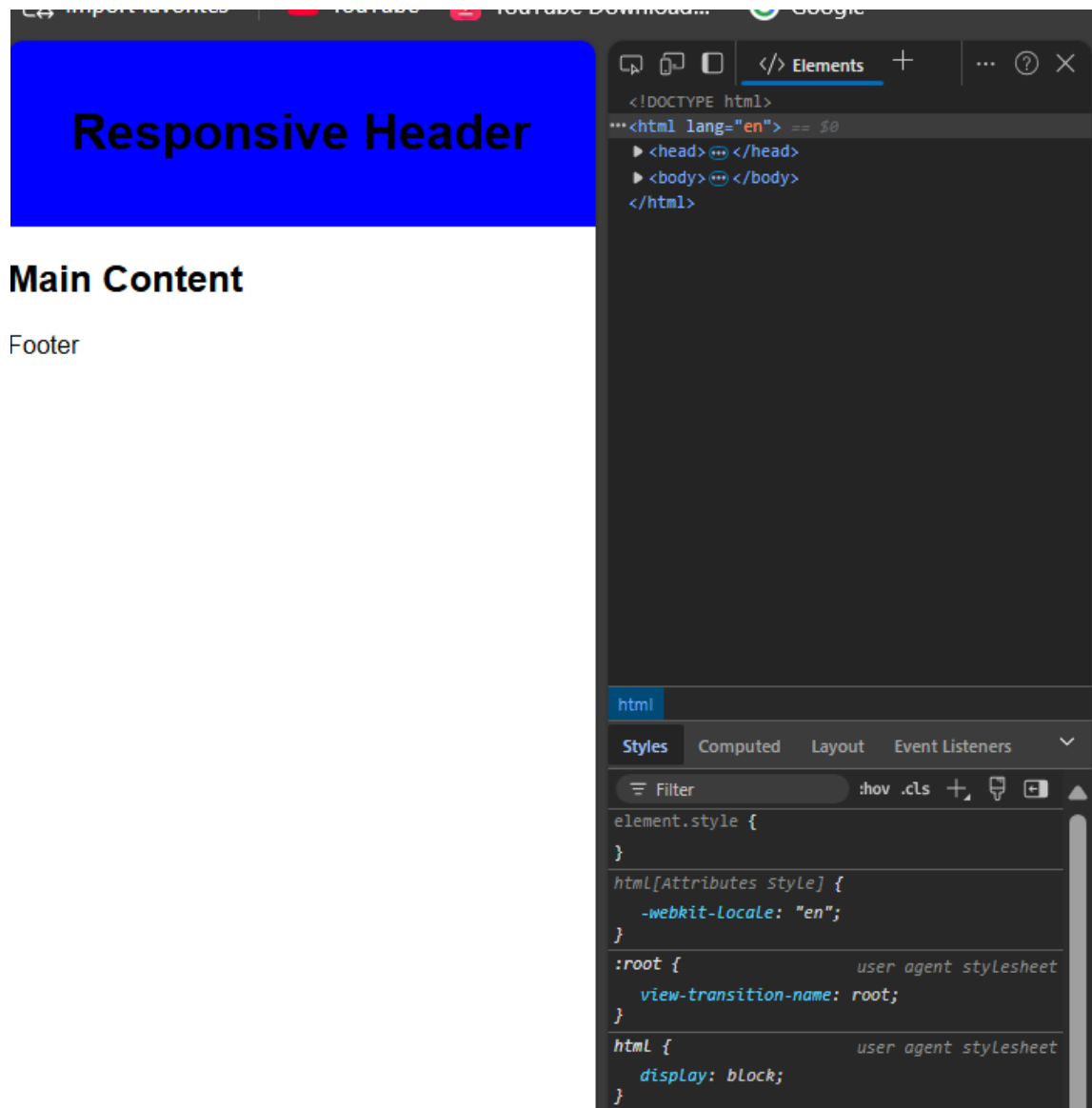
  <footer>
    <p>Footer</p>
  </footer>
</body>
</html>

```

Normal Output:



Responsive Output:



2.

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.

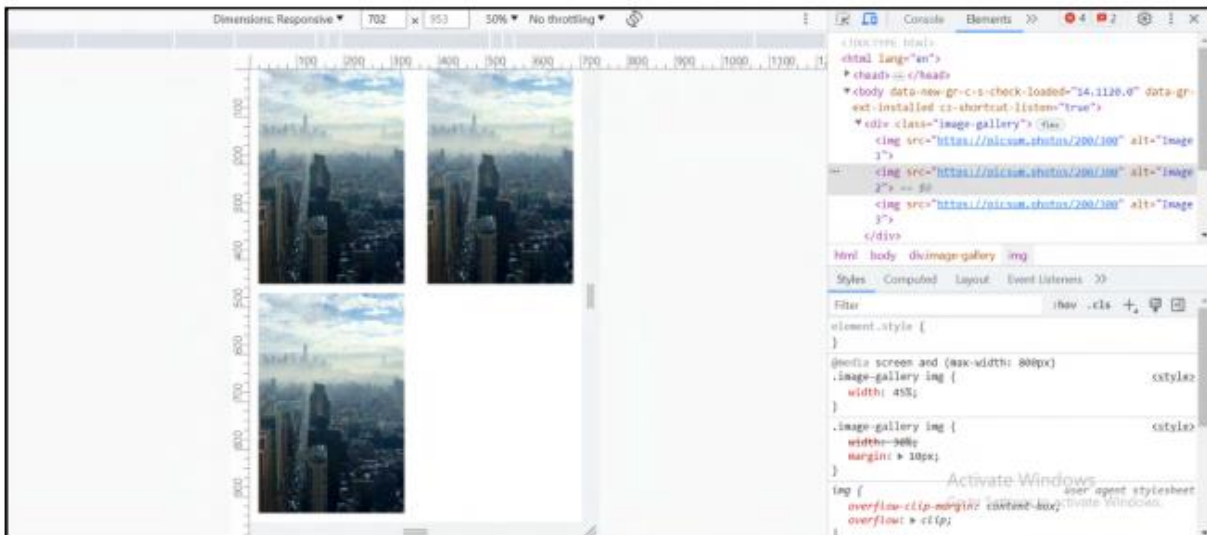
For image use this link ([Image Link](#))

Expected Output

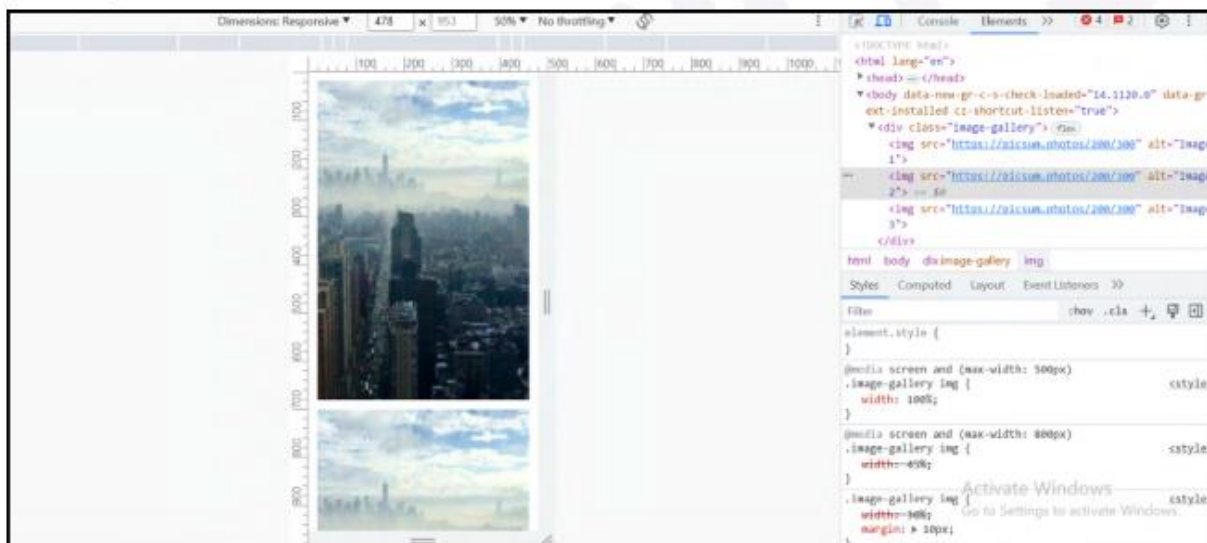
First Output



Second Output



Third Output



Ans: `<!DOCTYPE html>`

```
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Image Gallery</title>
</head>
<body>
```

```

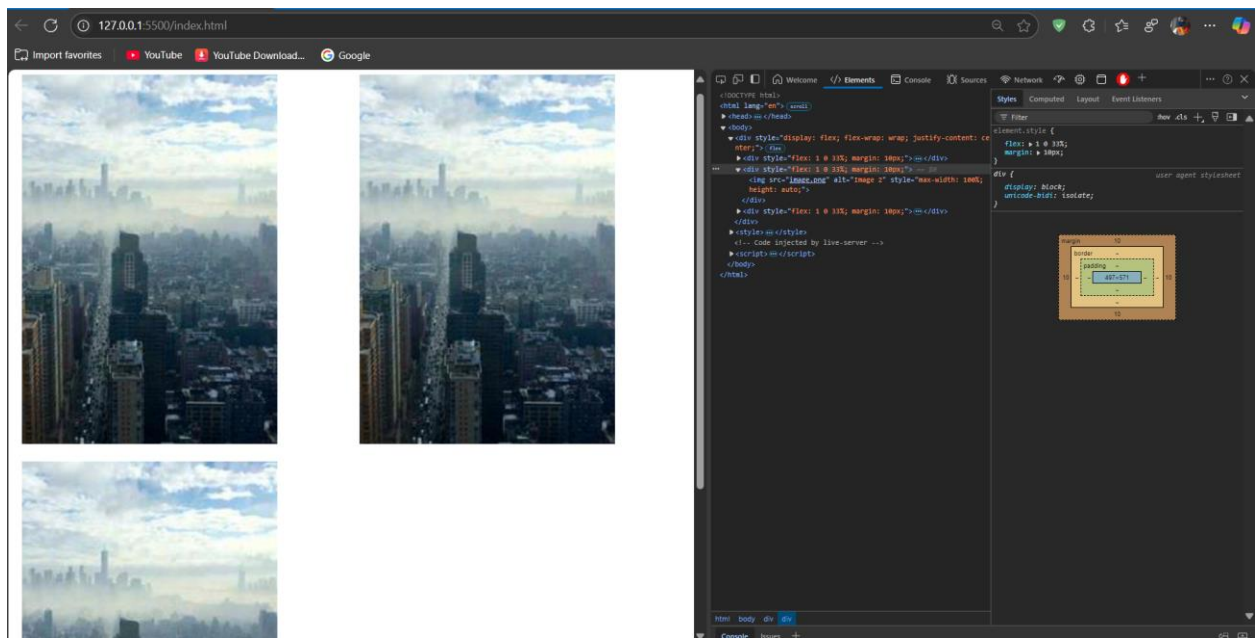
    <div style="display: flex; flex-wrap: wrap; justify-content: center;">
      <div style="flex: 1 0 33%; margin: 10px;">
        
      </div>
      <div style="flex: 1 0 33%; margin: 10px;">
        
      </div>
      <div style="flex: 1 0 33%; margin: 10px;">
        
      </div>
    </div>

    <style>
      @media (max-width: 800px) {
        div > div {
          flex: 1 0 100%;
        }
      }

      @media (max-width: 500px) {
        div > div {
          flex: 1 0 100%;
        }
      }
    </style>
  </body>
</html>

```

Ans:



3.

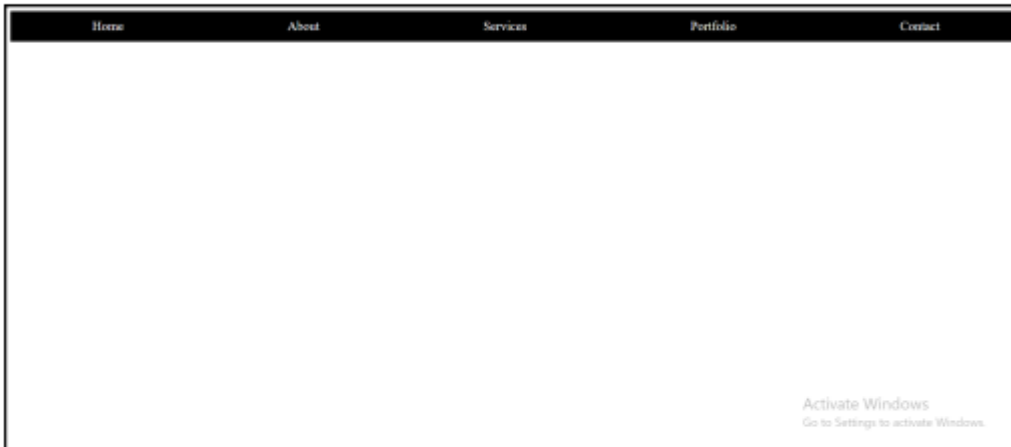
Task 3:

Problem Statement

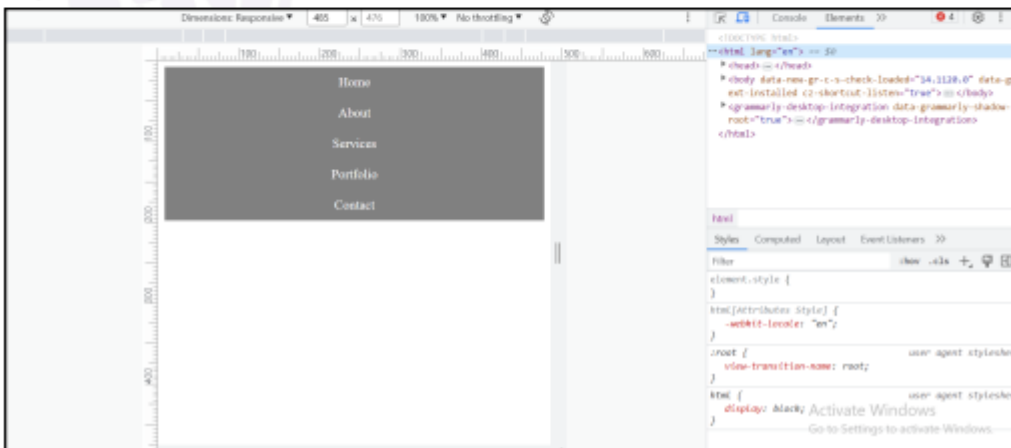
Create a navigation bar with five links. Apply a media query that converts the navigation into a vertical list for screens smaller than 768px and also changes the background color to gray.

Expected Outputs

Normal Output



Responsive Output



Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Navigation Bar</title>
  <style>
    nav {
      background-color: #333;
      padding: 10px;
```

```
}

nav ul {
  list-style: none;
  margin: 0;
  padding: 0;
  text-align: center;
}

nav li {
  display: inline;
  margin: 0 15px;
}

nav a {
  color: white;
  text-decoration: none;
  padding: 5px 10px;
}

@media (max-width: 768px) {
  nav {
    background-color: gray;
    padding: 10px;
  }

  nav ul {
    text-align: left;
  }

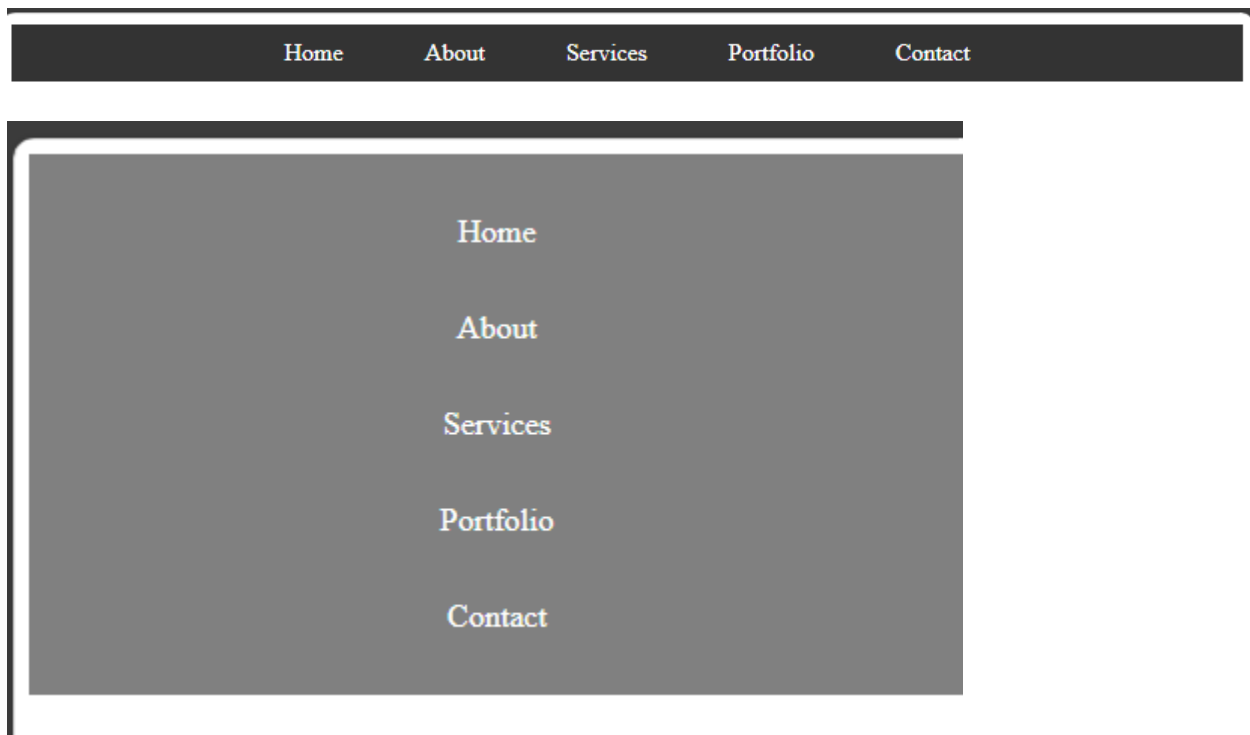
  nav li {
    display: block;
    margin: 10px 0;
  }

  nav a {
    display: flex;
    justify-content: center;
    padding: 10px;
    text-align: left;
  }
}

</style>
</head>
```

```
<body>
  <nav>
    <ul>
      <li><a href="#">Home</a></li>
      <li><a href="#">About</a></li>
      <li><a href="#">Services</a></li>
      <li><a href="#">Portfolio</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </nav>
</body>
</html>
```

Output:



Task 4:

Problem Statement

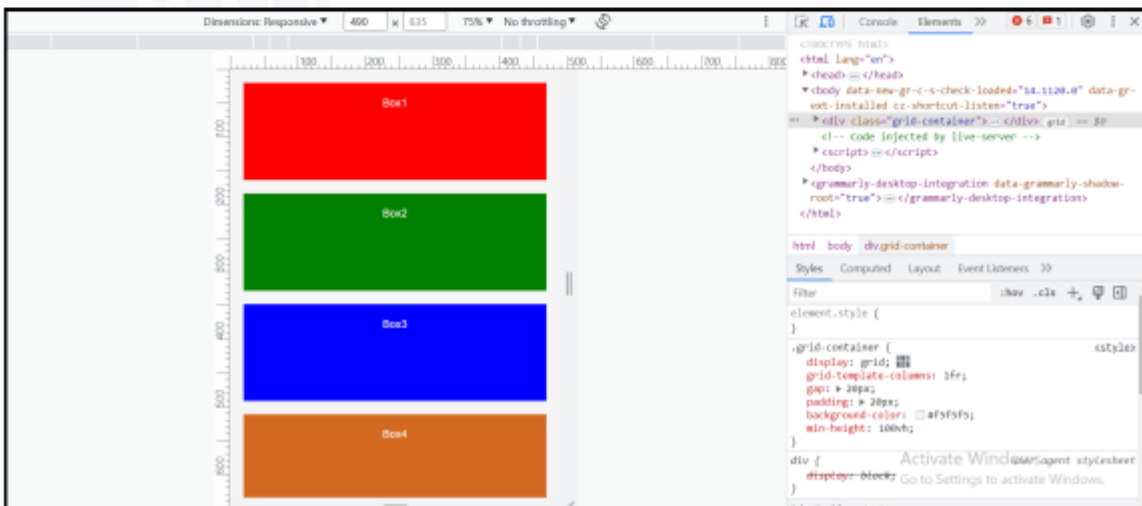
Create a grid based layout containing four grid items which arranges them as the screen size changes.

Expected Outputs

Normal Output



Responsive Output



Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Grid Layout</title>
<style>
  .grid-container {
    display: grid;
    grid-template-columns: repeat(3, 1fr);
    padding: 20px;
  }

  .grid-item {
    background-color: #f0f0f0;
    padding: 20px;
    text-align: center;
    border-radius: 5px;
    min-height: 100px;
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 20px;
    color: white;
  }

  .grid-item:nth-child(1) {
    background-color: red;
  }
  .grid-item:nth-child(2) {
    background-color: green;
  }
  .grid-item:nth-child(3) {
    background-color: blue;
  }
  .grid-item:nth-child(4) {
    background-color: orange;
    grid-column: span 2;
  }

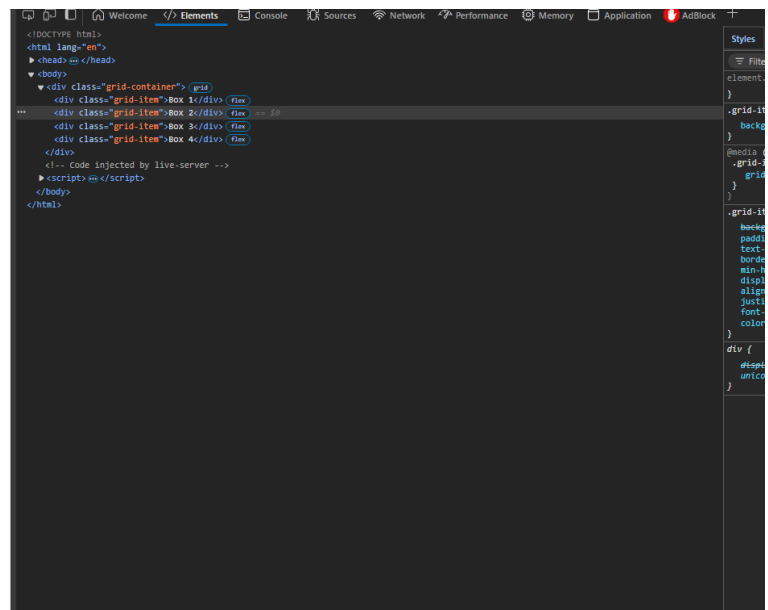
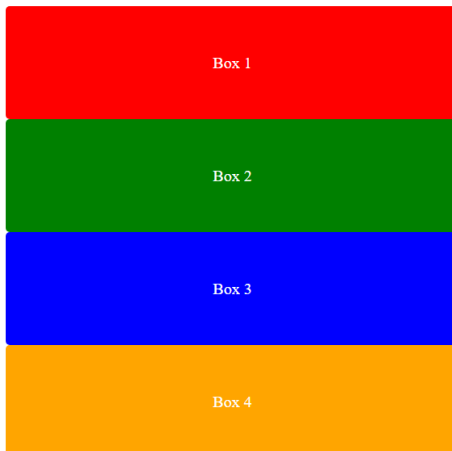
  @media (max-width: 768px) {
    .grid-container {
      grid-template-columns: 1fr;
    }
    .grid-item {
      grid-column: auto !important;
    }
  }
</style>
</head>
```

```
<body>

  <div class="grid-container">
    <div class="grid-item">Box 1</div>
    <div class="grid-item">Box 2</div>
    <div class="grid-item">Box 3</div>
    <div class="grid-item">Box 4</div>
  </div>

</body>
</html>
```

Output:



Task 5:

Problem Statement

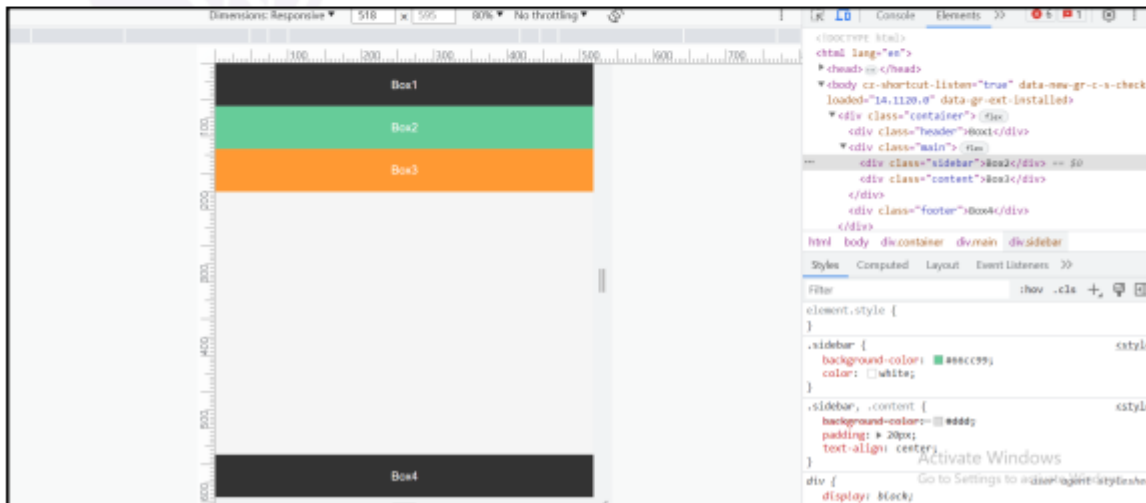
Create the below given layout using the flexbox in css, which should adapt itself on mobile screen as per given below output.

Expected Outputs

Normal Output



Responsive Output



Code:

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

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Flexbox Layout</title>
<style>
  body {
    font-family: sans-serif;
    margin: 0;
    padding: 0;
  }

  .container {
    display: flex;
    flex-direction: column;
    min-height: 100vh;
  }

  .box {
    display: flex;
    align-items: center;
    justify-content: center;
    text-align: center;
    color: white;
    font-size: 2em;
  }

  .box1 {
    flex: 0 0 auto;
    background-color: black;
  }

  .box2 {
    flex: 0 0 auto;
    background-color: lightseagreen;
  }

  .box3 {
    flex: 0 0 auto;
    background-color: orange;
  }

  .box4 {
    flex: 0 0 auto;
    background-color: black;
    margin-top: auto;
  }
}
```



```
@media (min-width: 1024px) {
  .container {
    flex-direction: row;
    justify-content: space-between;
  }

  .box1 {
    margin-left: 0;
    margin-right: auto;
  }

  .box4 {
    margin-left: auto;
    margin-right: 0;
  }

  .box2, .box3 {
    flex: 1;
    justify-content: center;
  }
}
</style>
</head>
<body>

  <div class="container">
    <div class="box box1">Box1</div>
    <div class="box box2">Box2</div>
    <div class="box box3">Box3</div>
    <div class="box box4">Box4</div>
  </div>

</body>
</html>
```

Output:

Box1
Box2
Box3

Box4
