

T1. Create a webpage that has a heading and a button. Write a script to change the background color and text when the button is clicked.

```
!<DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Change Background Color and Text</title>
<style>
    body {
        font-family: Arial, sans-serif;
        text-align: center;
        background-color: #f0f0f0;
    }
    h1 {
        color: #333;
    }
    button {
        padding: 10px 20px;
        font-size: 16px;
        background-color: #007bff;
        color: #fff;
        border: none;
        cursor: pointer;
    }
</style>
</head>
<body>

<h1 id="heading">Medimate</h1>
<button onclick="changeColorAndText()">Change Color and Text</button>

<script>
function changeColorAndText() {
    var heading = document.getElementById('heading');
    var button = document.querySelector('button');

    // Change background color
```

```

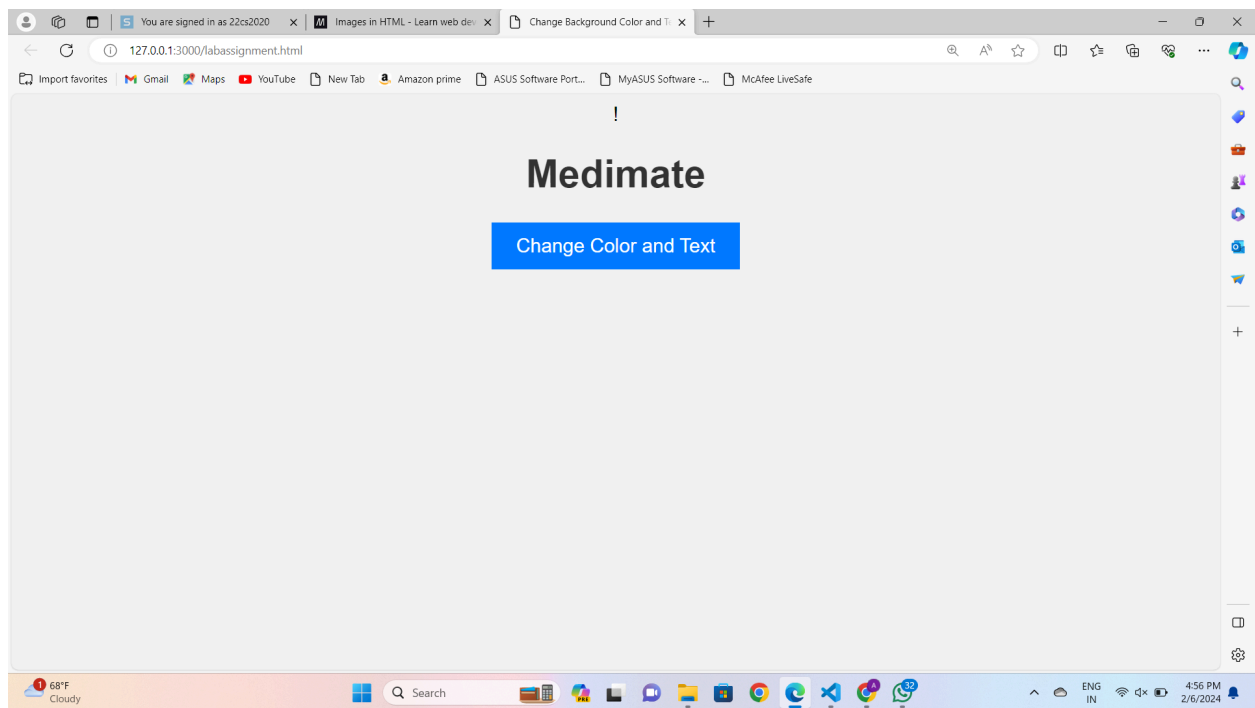
    var randomColor = '#' +
Math.floor(Math.random()*16777215).toString(16);
    document.body.style.backgroundColor = randomColor;

    // Change text of heading and button
    heading.textContent = "Healthcare";
    button.textContent = "Text Changed";
}
</script>

</body>
</html>

```

Output:



T2. For T1, experiment with the Javascript code by trying the following things:

- Make the background light green.
- Add a line that makes the text red.
- Change the fontSize to 48pt.
- Change the wording inside the <h1> tag in the Javascript.
- Add <button>Click Me</button> after the closing </h1>.

Code:

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Change Background Color and Text</title>
<style>
    body {
        font-family: Arial, sans-serif;
        text-align: center;
        background-color: lightgreen; /* changed to light green */
    }
    h1 {
        color: red; /* added to make the text red */
        font-size: 48pt; /* changed fontSize to 48pt */
    }
    button {
        padding: 10px 20px;
        font-size: 16px;
        background-color: #007bff;
        color: #fff;
        border: none;
        cursor: pointer;
    }
</style>
</head>
<body>

<h1 id="heading">Original Heading</h1>
<button onclick="changeColorAndText()">Change Color and Text</button>

<script>
function changeColorAndText() {
    var heading = document.getElementById('heading');
    var button = document.querySelector('button');

    // Change background color to light green
    document.body.style.backgroundColor = 'lightgreen';

    // Make the text red (already set in CSS)
```

```

// Change fontSize to 48pt (already set in CSS)

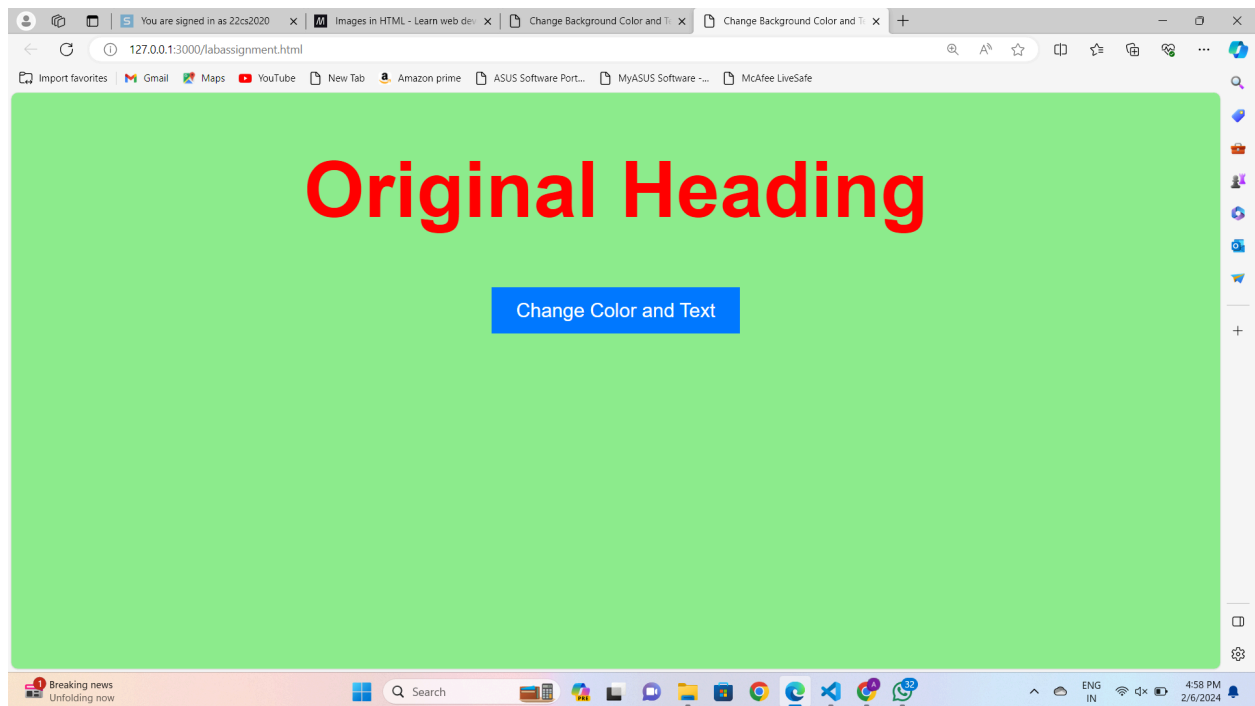
// Change text of heading and button
heading.textContent = "Modified Heading"; // changed wording inside
<h1> tag
button.textContent = "Text Changed";

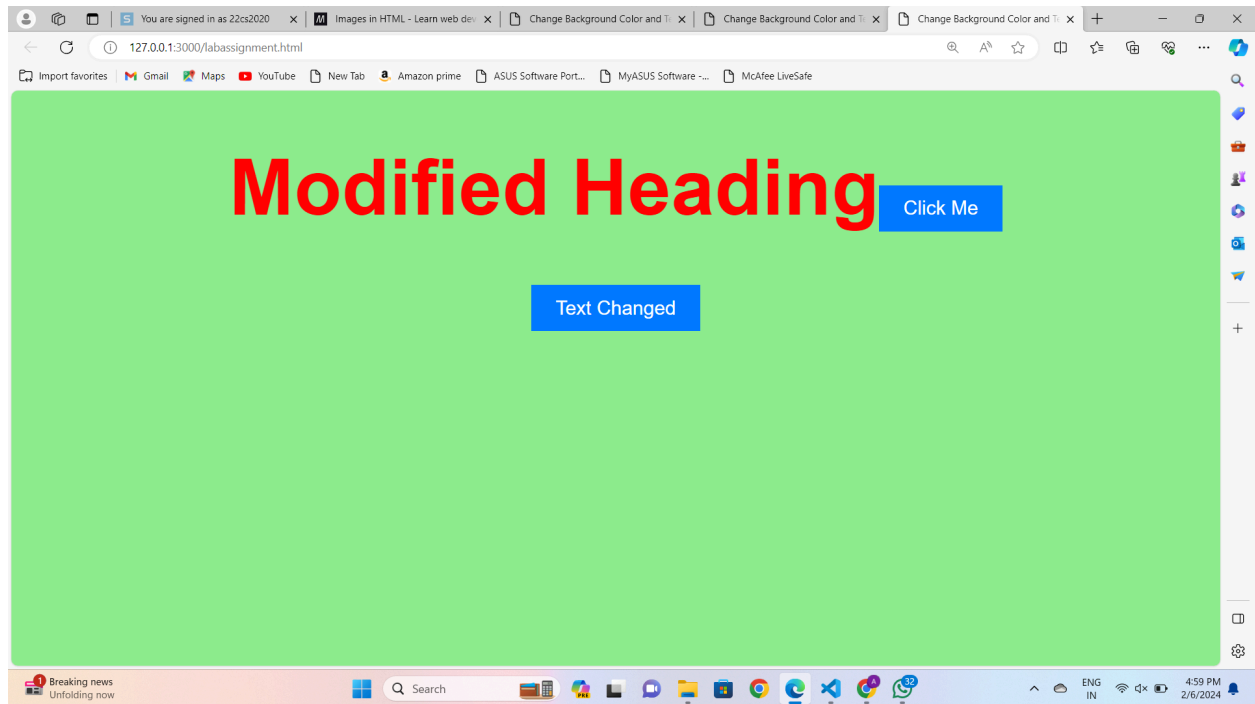
// Add <button>Click Me</button> after the closing </h1>
var newButton = document.createElement('button');
newButton.textContent = 'Click Me';
newButton.onclick = function() {
    alert('Button clicked!');
};
heading.appendChild(newButton);
}
</script>

</body>
</html>

```

Output:





T3. For T1, set the color of the background directly with the class attribute. Generate alerts before and after assigning the class attribute to document object.

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Change Background Color and Text</title>
<style>
  body {
    font-family: Arial, sans-serif;
    text-align: center;
    background-color: #f0f0f0;
  }
  .light-green-background {
    background-color: lightgreen;
  }
  h1 {
    color: #333;
  }
  button {
```

```
padding: 10px 20px;
font-size: 16px;
background-color: #007bff;
color: #fff;
border: none;
cursor: pointer;
}
</style>
</head>
<body>

<h1 id="heading">Original Heading</h1>
<button onclick="changeColorAndText()">Change Color and Text</button>

<script>
function changeColorAndText() {
    alert('Before assigning class attribute to document object');

    // Assign class attribute to change background color directly
    document.body.classList.add('light-green-background');

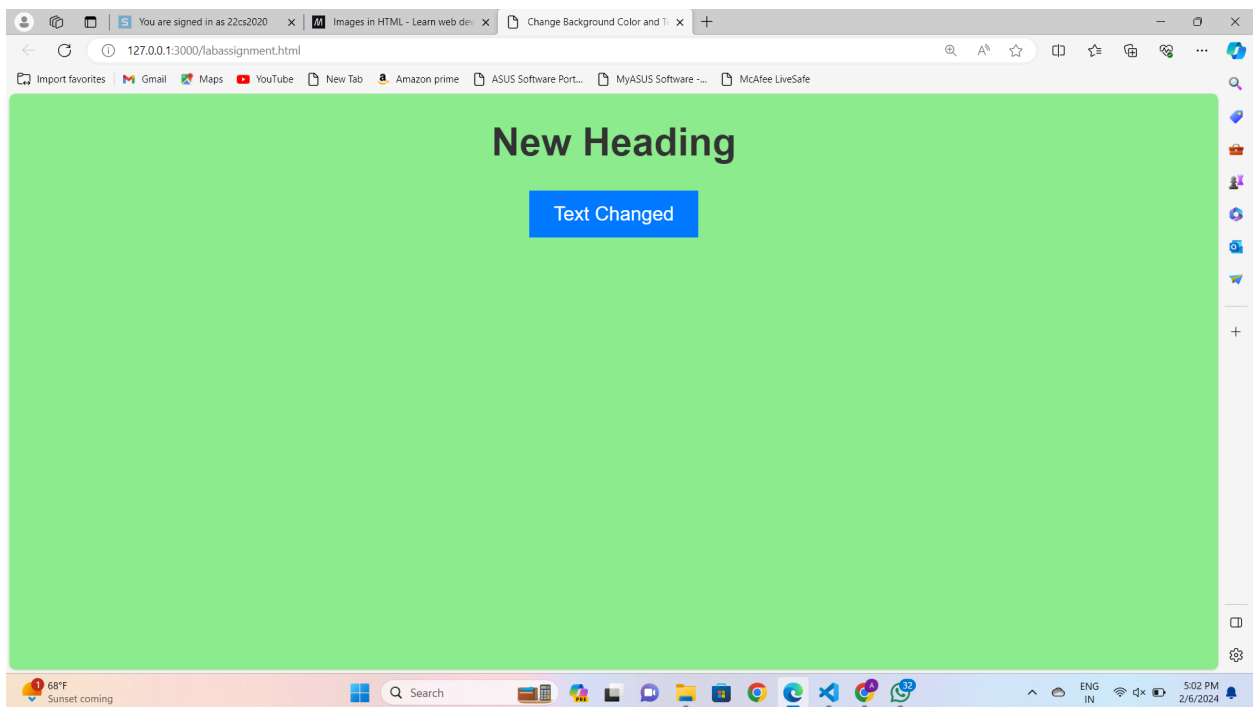
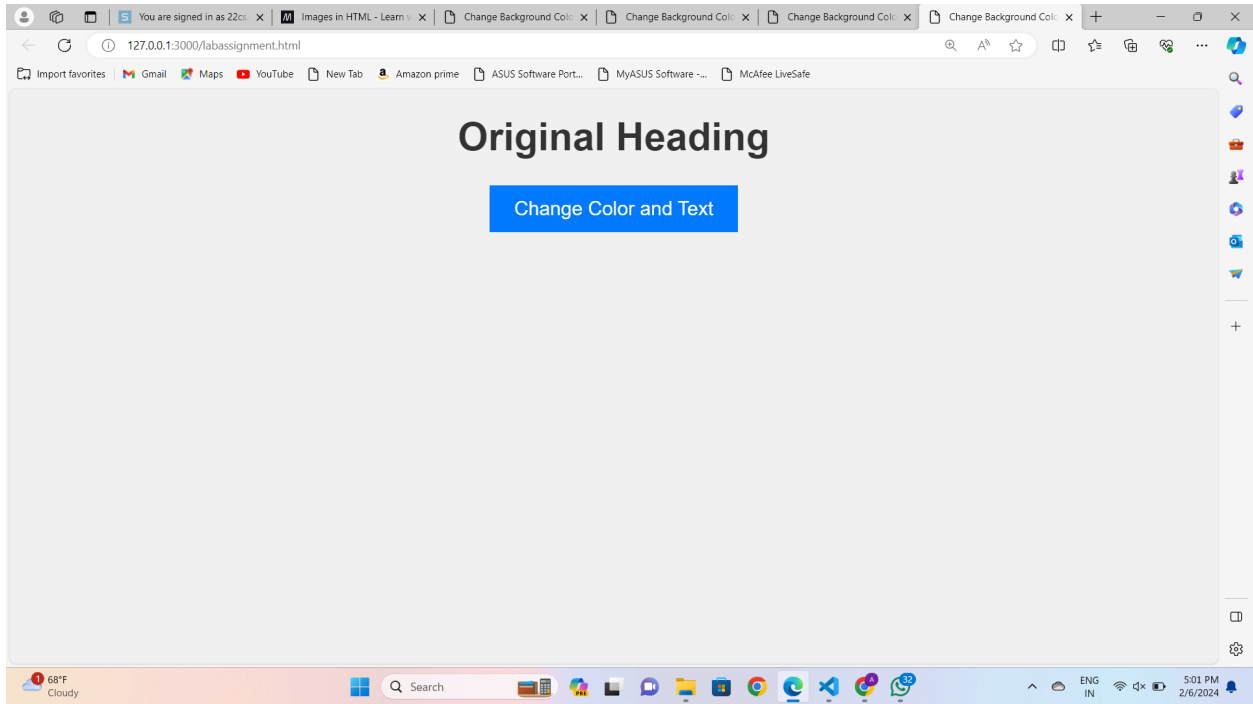
    alert('After assigning class attribute to document object');

    var heading = document.getElementById('heading');
    var button = document.querySelector('button');

    // Change text of heading and button
    heading.textContent = "New Heading";
    button.textContent = "Text Changed";
}
</script>

</body>
</html>
```

Output:



T4. Design a page having contents within and outside the main section. The page must contain a button that when clicked replaces the content of the main section with another content having different style.

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Page with Content Replacement</title>
<style>
  body {
    font-family: Arial, sans-serif;
    margin: 0;
    padding: 0;
  }
  header {
    background-color: #007bff;
    color: #fff;
    padding: 10px;
    text-align: center;
  }
  main {
    padding: 20px;
  }
  .original-content {
    font-size: 18px;
    line-height: 1.6;
  }
  .new-content {
    font-size: 20px;
    line-height: 1.8;
    color: #f00;
    font-weight: bold;
  }
  button {
    padding: 10px 20px;
    font-size: 16px;
    background-color: #007bff;
    color: #fff;
    border: none;
    cursor: pointer;
  }
</style>
```



```
</head>
<body>

<header>
  <h1>Page with Content Replacement</h1>
</header>

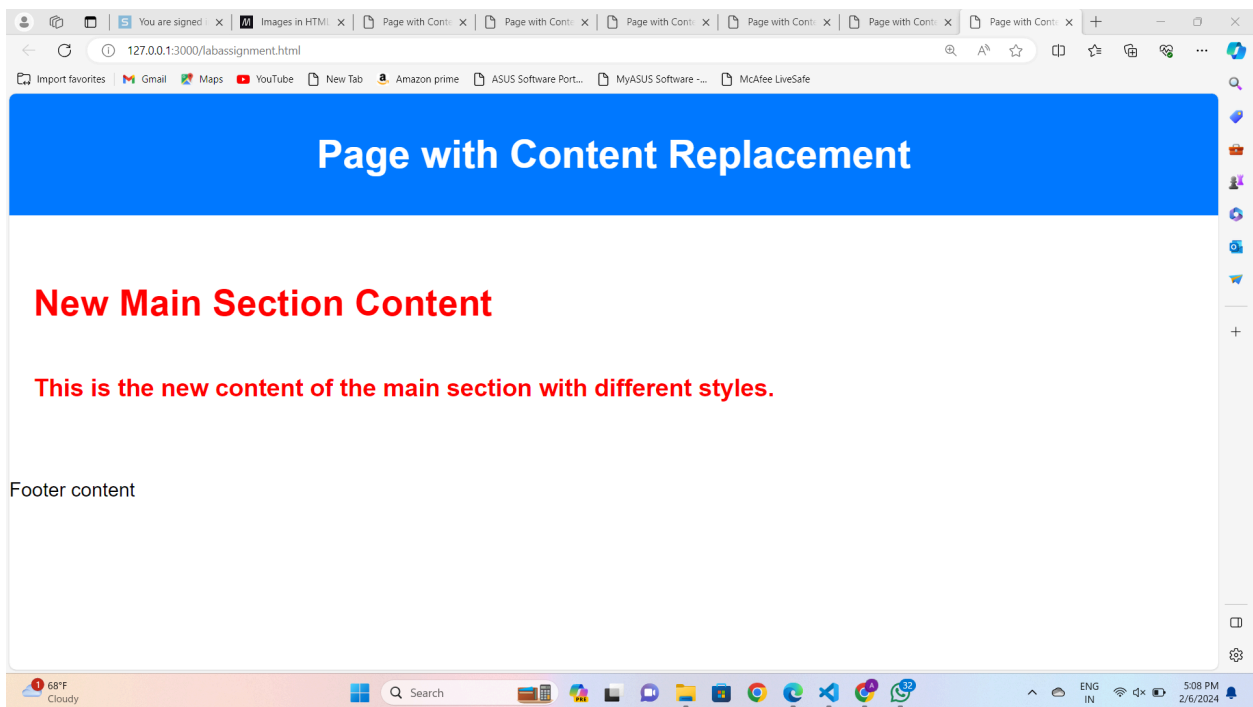
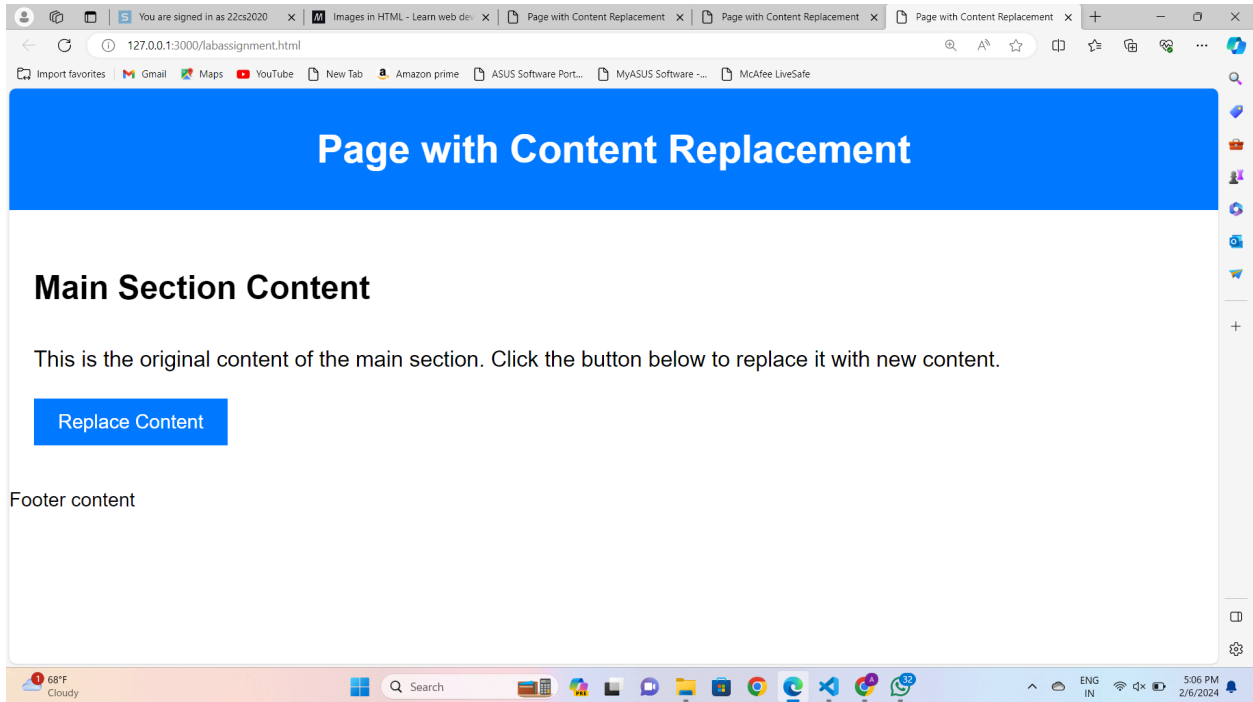
<main class="original-content">
  <h2>Main Section Content</h2>
  <p>This is the original content of the main section. Click the button
below to replace it with new content.</p>
  <button onclick="replaceContent()">Replace Content</button>
</main>

<footer>
  <p>Footer content</p>
</footer>

<script>
function replaceContent() {
  var mainSection = document.querySelector('main');
  mainSection.classList.remove('original-content');
  mainSection.classList.add('new-content');
  mainSection.innerHTML = '<h2>New Main Section Content</h2><p>This is
the new content of the main section with different styles.</p>';
}
</script>

</body>
</html>
```

Output:



T5. Consider a h1 heading lying outside the main section in the page. Try changing the color of h1 in at least 3 different ways.

Using Inline Style Attribute: You can use the `style` attribute directly on the `h1` element to specify its color.

Code:

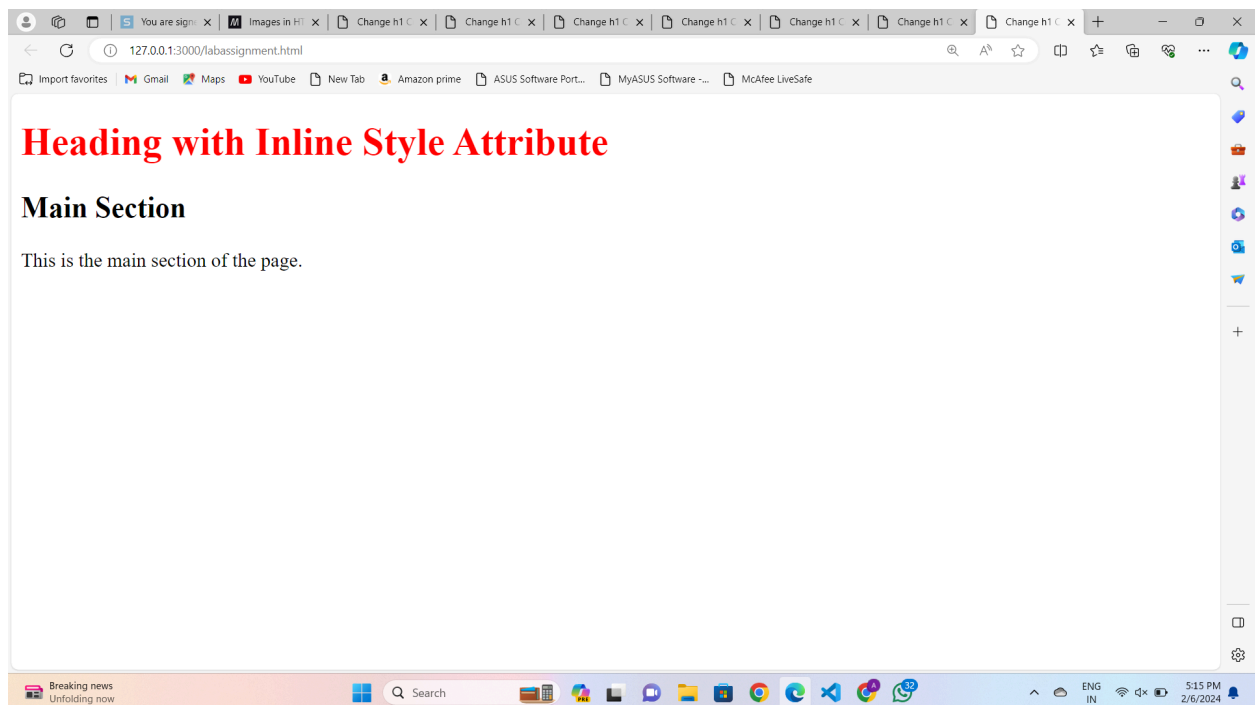
```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Change h1 Color</title>
</head>
<body>

<h1 style="color: red;">Heading with Inline Style Attribute</h1>

<main>
  <h2>Main Section</h2>
  <p>This is the main section of the page.</p>
</main>

</body>
</html>
```

Output:



Using Internal CSS: You can define a style block within the `head` section of your HTML document and target the `h1` element to specify its color.

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Change h1 Color</title>
<style>
  h1 {
    color: green;
  }
</style>
</head>
<body>

<h1>Heading with Internal CSS</h1>

<main>
  <h2>Main Section</h2>
  <p>This is the main section of the page.</p>
</main>

</body>
</html>
```

Output:



Using External CSS: You can define a separate CSS file and link it to your HTML document to specify the color of the `h1` element.

Code:

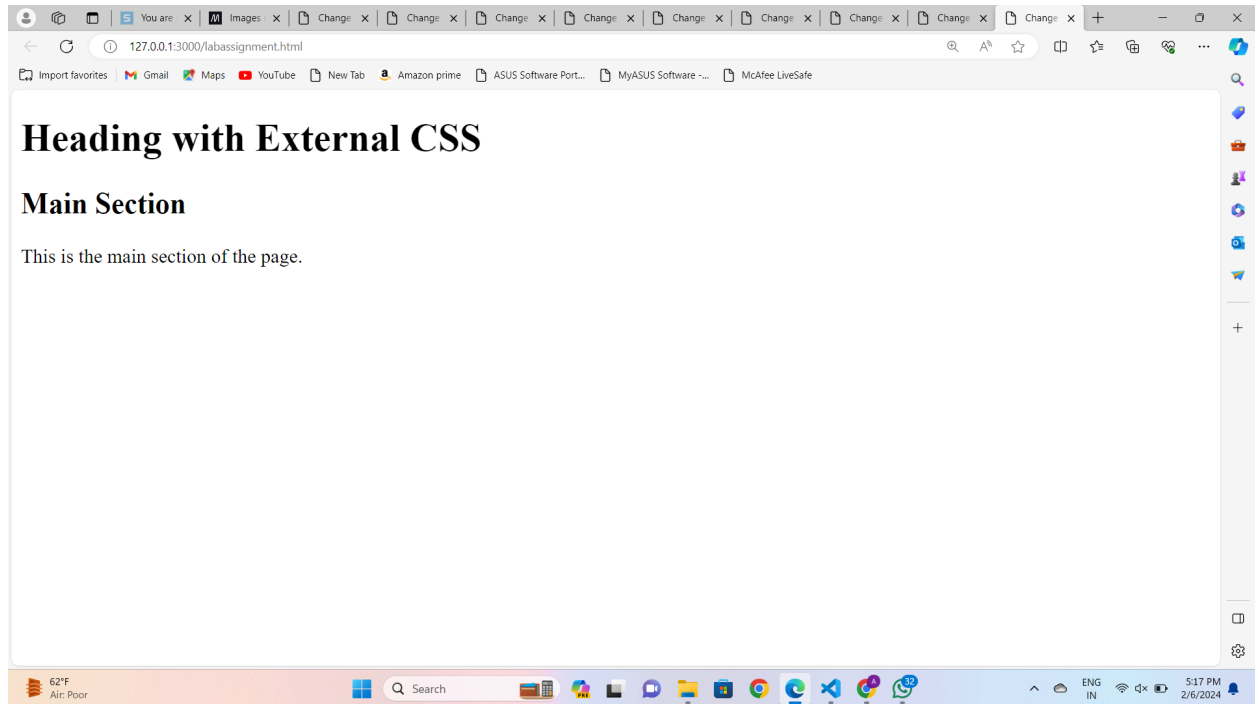
```
<!-- index.html -->
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Change h1 Color</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>

<h1>Heading with External CSS</h1>

<main>
  <h2>Main Section</h2>
  <p>This is the main section of the page.</p>
</main>
```

```
</body>
</html>
```

Output:



T6. Design a webpage using grid layout that displays the following:

The first row shows icons of various web browsers and the second row shows the latest version number.

Code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Web Browser Versions</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 0;
```

```

    box-sizing: border-box;
}

.grid-container {
    display: grid;
    grid-template-rows: auto auto; /* Two rows with automatic height */
    grid-template-columns: repeat(5, 1fr); /* Five columns */
    gap: 10px;
    padding: 20px;
}

.browser-icon {
    width: 100%;
    height: auto;
    text-align: center;
}

.browser-icon img {
    width: 100px;
}

.version-number {
    text-align: center;
    background-color: #f2f2f2; /* Light grey background color */
    padding: 10px; /* Add some padding for better visibility */
}
</style>
</head>
<body>

<div class="grid-container">
    <!-- Icons of various web browsers -->
    <div class="browser-icon"></div>
    <div class="browser-icon"></div>
    <div class="browser-icon"></div>
    <div class="browser-icon"></div>
    <div class="browser-icon"></div>
    <!-- Add more browser icons as needed -->

```

```
<!-- Latest version numbers -->
<div class="version-number">Chrome<br>Version 100.0</div>
<div class="version-number">Firefox<br>Version 99.0</div>
<div class="version-number">Edge<br>Version 20.0</div>
<div class="version-number">Safari<br>Version 15.0</div>
<div class="version-number">Opera<br>Version 75.0</div>
<!-- Add more version numbers as needed -->
</div>

</body>
</html>
```

Output:

