## Context: You are an accessibility expert reviewing HTML for accessibility issues. Analyze the following code in the context of WCAG (Web Content Accessibility Guidelines) 2.2, listing any issues that may violate these guidelines. You will only be analyzing the HTML, ignoring any included or missing styles and scripts.

Code:

```<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<style>

body {

margin: 0;

padding: 0;

font-family: 'Arial', sans-serif;

background: url('background.jpg') no-repeat center center fixed;

background-size: cover;

color: #fff;

}

header {

background: #333;

padding: 10px;

text-align: center;

}

nav {

display: flex;

justify-content: center;

background: #444;

padding: 10px;

}

nav a {

color: #fff;

text-decoration: none;

margin: 0 20px;

font-size: 18px;

}

section {

padding: 20px;

}

h2 {

color: #fff; /\* Text color: white \*/

background-color: rgba(0, 0, 0, 0.7); /\* Semi-transparent black background for better contrast \*/

padding: 10px; /\* Add padding for better appearance \*/

}

form {

display: flex;

flex-direction: column;

max-width: 300px;

margin: auto;

}

label {

margin-bottom: 8px;

color: #fff; /\* Set label text color to white for better contrast \*/

background-color: rgba(0, 0, 0, 0.7); /\* Semi-transparent black background for better contrast \*/

padding: 8px; /\* Add padding for better appearance \*/

}

input, select {

padding: 10px;

margin-bottom: 15px;

}

button {

padding: 10px;

background: #4CAF50;

color: #fff;

border: none;

cursor: pointer;

}

button:hover {

background: #45a049;

}

</style>

</head>

<body>

<header>

<marquee>Ticket Booking System</marquee>

</header>

<nav>

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

<a href="#">Movies</a>

<a href="#">Events</a>

<a href="#">Contact</a>

</nav>

<section>

<h2>Book Your Tickets</h2>

<form id="bookingForm">

<label for="event"></label>

<select id="event" name="event">

<option value="movie">Movie</option>

<option value="concert">Concert</option>

<!-- Add more options as needed -->

</select>

<label for="quantity">

Number of Tickets:

</label>

<input type="number" alt="quantity" id="quantity" name="quantity" min="1" required>

<label for="name" style="color: #fff; background-color: rgba(0, 0, 0, 0.7);">

Your Name:

</label>

<input type="text" id="name" name="name" required>

<button type="button" onclick="submitForm()" style="background-color: #45a049;"> <!-- Adjusted background color for better contrast -->

Book Now

</button>

</form>

<div id="confirmation" style="display:none;">

<p id="confirmationText">Your tickets have been booked. Thank you!</p>

</div>

</section>

<script>

function submitForm() {

// Perform any additional processing here (e.g., sending data to a server)

// For now, just show a confirmation message

document.getElementById('bookingForm').style.display = 'none';

document.getElementById('confirmation').style.display = 'block';

}

</script>

</body>

</html>

```

Response format:

Snippet with error: …

Location of error: …

WCAG Guideline violated: …

Detection Algorithm: …

Explanation of Error: …

### Snippet with error:

```html

<button type="button" onclick="submitForm()" style="background-color: #45a049;">

Book Now

</button>

```

### Location of error:

This issue is located within the `<button>` element inside the `<form>` tag in the `<section>` of the HTML document.

### WCAG Guideline violated:

\*\*Perceivable:\*\* 1.3 Adaptable (Sufficient): Ensure that pages are designed so they are accessible to a wide range of user agents, including assistive technologies. This includes providing alternatives for time-based media and ensuring Web content and functionality are available from a variety of devices.

### Detection Algorithm:

The algorithm checks for the presence of interactive elements like buttons that rely solely on JavaScript for their functionality without providing alternative mechanisms for users who cannot interact with JavaScript-based interfaces.

### Explanation of Error:

The button uses inline JavaScript (`onclick`) to handle its click event, which submits the form. However, this approach does not provide a fallback mechanism for users who might have JavaScript disabled or use assistive technologies that cannot execute JavaScript. The button lacks a `type` attribute specifying its role, such as `type="submit"` for submitting forms, which could also contribute to confusion about its purpose.

To address this issue, consider adding a `type="submit"` attribute to the button and providing a visible indication that the button submits the form, even if JavaScript is disabled. Additionally, ensure that the form submission process works without relying solely on JavaScript for critical interactions.

Citations:

## Are there any more issues? List all that you find.

Upon further analysis of the provided HTML code, additional accessibility concerns under WCAG 2.2 guidelines are identified:

### Snippet with error:

```html

<label for="event"></label>

<select id="event" name="event">

<option value="movie">Movie</option>

<option value="concert">Concert</option>

</select>

```

### Location of error:

This issue is located within the `<label>` and `<select>` elements inside the `<form>` tag in the `<section>` of the HTML document.

### WCAG Guideline violated:

\*\*Understandable:\*\* 1.4.6 Assistive Technologies (A): Where animations, whether whole pages or fragments, continuously loop, the function shall be interruptible. This is not applicable in this snippet, but it highlights the importance of considering animations and interactivity in the context of accessibility.

### Detection Algorithm:

The algorithm looks for `<label>` elements that are associated with form controls but lack descriptive text content. This can lead to confusion for screen reader users who rely on labels to understand the purpose of form controls.

### Explanation of Error:

The `<label>` element for the `<select>` dropdown is empty, which means screen readers will announce the dropdown without any preceding label, potentially confusing users about its purpose. To resolve this, the `<label>` should contain descriptive text that clearly indicates the purpose of the dropdown.

### Snippet with error:

```html

<input type="number" alt="quantity" id="quantity" name="quantity" min="1" required>

```

### Location of error:

This issue is located within the `<input>` element inside the `<form>` tag in the `<section>` of the HTML document.

### WCAG Guideline violated:

\*\*Operable:\*\* 1.3.1 Info and Relationships (A): Information, structure, and relationships conveyed through sequencing or markup are presented in a way that produces a correct and perceivable meaning when linearized (e.g., as in a printed document).

### Detection Algorithm:

The algorithm identifies incorrect usage of the `alt` attribute on an `<input>` element of type `number`. The `alt` attribute is meant for images, not inputs, and its presence on an input field is misleading and unnecessary.

### Explanation of Error:

The `alt` attribute is incorrectly used on an `<input>` element of type `number`, which is not an image. This misuse can confuse developers and tools checking for accessibility compliance. The `alt` attribute should be removed from this context.

### Corrected Code:

```html

<!-- Corrected label for the select element -->

<label for="event">Event Type:</label>

<select id="event" name="event">

<option value="movie">Movie</option>

<option value="concert">Concert</option>

</select>

<!-- Removed alt attribute from the input element -->

<input type="number" id="quantity" name="quantity" min="1" required>

```

These corrections address the identified accessibility issues by ensuring proper labeling for form controls and removing the misuse of the `alt` attribute, thereby improving the overall accessibility of the webpage according to WCAG 2.2 guidelines.

Citations:

[1] https://www.w3.org/TR/WCAG22/

[2] https://www.accessibility.works/blog/wcag-2-2-guide/

[3] https://equalizedigital.com/wcag-2-2-explained-how-to-test-for-it/

[4] https://www.audioeye.com/compliance/wcag/

[5] https://www.w3.org/WAI/test-evaluate/tools/list/

[6] https://abilitynet.org.uk/news-blogs/wcag-22-what-you-need-know

[7] https://github.com/w3c/wcag/issues/2820

[8] https://www.siteimprove.com/blog/wcag-2-2-is-here-what-does-it-mean-for-you/

[9] https://www.audioeye.com/post/whats-new-with-wcag-2-2/

[10] https://adasitecompliance.com/top-10-things-know-about-wcag-22/

## Are there any more issues? List all that you find.

After a thorough review of the provided HTML code against the WCAG 2.2 guidelines, additional accessibility issues have been identified:

### Snippet with error:

```html

<header>

<marquee>Ticket Booking System</marquee>

</header>

```

### Location of error:

This issue is located within the `<header>` element at the beginning of the `<body>` section of the HTML document.

### WCAG Guideline violated:

\*\*Understandable:\*\* 1.4.5 Contrast (Minimum). A contrast ratio of at least 4.5:1 is required between text (and images of text) and its background for normal text and 3:1 for large text, with exceptions for small text.

### Detection Algorithm:

The algorithm checks for the use of the `<marquee>` element, which is deprecated and not recommended for use in modern web development. It also assesses the contrast ratio between the text color and the background color to ensure it meets the minimum requirements for readability and accessibility.

### Explanation of Error:

The `<marquee>` element is deprecated and not supported in HTML5. It is replaced by CSS animations or JavaScript for creating scrolling text effects. Moreover, the contrast between the marquee text ("Ticket Booking System") and its background is not explicitly defined, which could lead to poor visibility for users with visual impairments.

### Snippet with error:

```html

<div id="confirmation" style="display:none;">

<p id="confirmationText">Your tickets have been booked. Thank you!</p>

</div>

```

### Location of error:

This issue is located within the `<div>` element with the ID "confirmation" towards the end of the `<body>` section of the HTML document.

### WCAG Guideline violated:

\*\*Operable:\*\* 2.4.7 Focus Visible (A). Any keyboard operable user interface component shall present focus state when selected by keyboard navigation unless the component does not receive focus through keyboard interaction.

### Detection Algorithm:

The algorithm checks for the visibility of focus states for interactive elements, ensuring that users navigating with keyboards can easily identify which element is currently focused.

### Explanation of Error:

The `<div>` element with the ID "confirmation" is styled to be hidden by default (`display:none`). While this is not inherently an accessibility issue, the problem arises when attempting to manage focus after showing this element dynamically (e.g., after form submission). Since the element is initially hidden, it does not naturally participate in the tab order, potentially causing confusion for keyboard users when trying to navigate to it.

### Corrected Code:

To address the identified issues, consider replacing the `<marquee>` element with a CSS animation or JavaScript for a modern, accessible implementation. Ensure that dynamic elements like the confirmation message are properly integrated into the tab order and have visible focus indicators when shown. Here's an example correction for the `<div>` element:

```html

<style>

.hidden { display: none; }

.visible:focus { outline: 2px solid blue; } /\* Example focus indicator \*/

</style>

<div id="confirmation" class="hidden">

<p id="confirmationText">Your tickets have been booked. Thank you!</p>

</div>

<script>

function submitForm() {

document.getElementById('bookingForm').classList.add('hidden');

document.getElementById('confirmation').classList.remove('hidden');

document.getElementById('confirmation').focus(); // Set focus to the confirmation message

}

</script>

```

This correction ensures that the confirmation message is accessible to keyboard users by managing its visibility and focus appropriately.

Citations:

[1] https://www.w3.org/TR/WCAG22/

[2] https://www.accessibility.works/blog/wcag-2-2-guide/

[3] https://www.w3.org/WAI/WCAG22/Understanding/accessible-authentication-minimum.html

[4] https://w3c.github.io/wcag/techniques/

[5] https://github.com/w3c/wcag/issues/2820

[6] https://www.milespartnership.com/how-we-think/article/tackling-most-common-web-accessibility-issues-2024

[7] https://adasitecompliance.com/wcag-2-2-decoding-latest-web-accessibility-guidelines/

[8] https://tetralogical.com/blog/2023/10/05/whats-new-wcag-2.2/

[9] https://www.w3.org/WAI/test-evaluate/tools/list/

[10] https://accessibility.psu.edu/guidelines/wcaglist/