

Name - Vivek Pahilwan PRN - 22320030 Roll - 331071	Name - Manasi Hire PRN - 22320015 Roll - 331070
--	---

Assignment 1 - Accessibility Audit: Evaluate a public website or app for accessibility compliance and propose improvements.

Introduction

In this assignment, we are conducting an **Accessibility Audit** of the **IRCTC website** (<https://www.irctc.co.in>), which serves as the official platform for the **Indian Railway Catering and Tourism Corporation (IRCTC)**. The IRCTC website is a critical public resource, enabling users to book train tickets, plan travel itineraries, check train schedules, and access various tourism and catering services.

With millions of users accessing the platform daily, it caters to a diverse audience, including people with varying levels of digital literacy and users with disabilities. Therefore, ensuring the website is accessible, inclusive, and compliant with **Web Content Accessibility Guidelines (WCAG)** is essential to providing an equitable user experience.

This audit evaluates the IRCTC website's adherence to accessibility standards using **Lighthouse**, a tool integrated into Chrome Developer Tools. It identifies accessibility issues, such as ARIA roles, contrast ratios, missing labels, and keyboard navigation, and proposes improvements to enhance the usability of the website for all users, including those relying on assistive technologies like screen readers.

By addressing these issues, the website can ensure inclusivity and align with India's commitment to **digital accessibility**, as outlined in the **Rights of Persons with Disabilities Act, 2016**, and global web accessibility standards.

The image displays a screenshot of the IRCTC website's 'BOOK TICKET' interface on the left and a Lighthouse accessibility audit report on the right.

IRCTC Website Interface:

- BOOK TICKET** section with fields for From, To, Date (19/01/2025), All Classes, and GENERAL.
- Search button.
- Easy Booking on AskDISHA button.
- Customer Care Numbers: 14646/08044647999 /08035734999 (Language: Hindi & English).
- Advertisement for ₹5 Lakh Free Treatment for Senior Citizens (70 years & above).
- Chatbot interface with 'Type here to chat...' and 'Book Train Ticket' button.

Lighthouse Accessibility Audit Report:

- Score: 60
- Section: Accessibility
- Message: There were issues affecting this run of Lighthouse: There may be stored data affecting loading performance in this location: IndexedDB. Audit this page in an incognito window to prevent those resources from affecting your scores.
- Issues listed under ARIA:

Issue
▲ [aria-+] attributes do not match their roles
▲ [aria-+] attributes do not have valid values
▲ button, link, and menuitem elements do not have accessible names.
▲ [aria-hidden="true"] elements contain focusable descendants
▲ Elements use prohibited ARIA attributes
▲ Uses ARIA roles on incompatible elements
○ ARIA IDs are unique



There were issues affecting this run of Lighthouse:

- There may be stored data affecting loading performance in this location: IndexedDB. Audit this page in an incognito window to prevent those resources from affecting your scores.



Accessibility

These checks highlight opportunities to [improve the accessibility of your web app](#). Automatic detection can only detect a subset of issues and does not guarantee the accessibility of your web app, so [manual testing](#) is also encouraged.

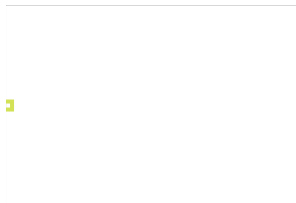
ARIA

▲ [aria-*] attributes do not match their roles



Each ARIA role supports a specific subset of aria-* attributes. Mismatching these invalidates the aria-* attributes. [Learn how to match ARIA attributes to their roles](#).

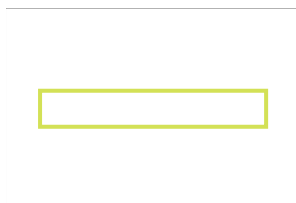
Failing Elements



div.ng-tns-c47-5.ng-trigger.ng-trigger-panelState.ui-sidebar.ui-widget.ui-widget-content.ui-shadow



input.ng-tns-c57-8.ui-inputtext.ui-widget.ui-state-default.ui-corner-all.ui-autocomplete-input.ng-star-inserted



input.ng-tns-c57-9.ui-inputtext.ui-widget.ui-state-default.ui-corner-all.ui-autocomplete-input.ng-star-inserted

▲ [aria-*] attributes do not have valid values



Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid values. [Learn more about valid values for ARIA attributes.](#)

Failing Elements



input.ng-tns-c57-8.ui-inputtext.ui-widget.ui-state-default.ui-corner-all.ui-autocomplete-input.ng-star-inserted



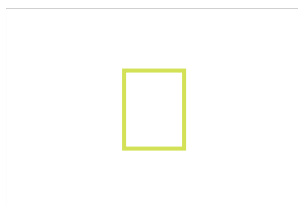
input.ng-tns-c57-9.ui-inputtext.ui-widget.ui-state-default.ui-corner-all.ui-autocomplete-input.ng-star-inserted

▲ button, link, and menuitem elements do not have accessible names.

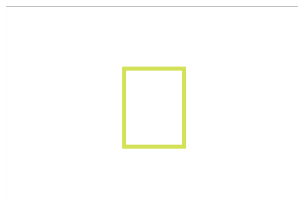


When an element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn how to make command elements more accessible.](#)

Failing Elements



div.ui-dropdown-trigger.ui-state-default.ui-corner-right.ng-tns-c65-11



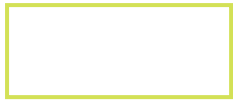
div.ui-dropdown-trigger.ui-state-default.ui-corner-right.ng-tns-c65-12

▲ [aria-hidden="true"] elements contain focusable descendents



Focusable descendents within an [aria-hidden="true"] element prevent those interactive elements from being available to users of assistive technologies like screen readers. [Learn how aria-hidden affects focusable elements.](#)

Failing Elements



div#v-irctc-co-in-0



▲ Elements use prohibited ARIA attributes



Using ARIA attributes in roles where they are prohibited can mean that important information is not communicated to users of assistive technologies. [Learn more about prohibited ARIA roles.](#)

Failing Elements



p-autocomplete#origin.form-group.ng-tns-c57-8.ng-pristine.ng-invalid.ng-touched



a.col-xs-push-11.col-sm-push-6.swap



p-autocomplete#destination.form-group.ng-tns-c57-9.ng-untouched.ng-pristine.ng-invalid

▲ Uses ARIA roles on incompatible elements



Many HTML elements can only be assigned certain ARIA roles. Using ARIA roles where they are not allowed can interfere with the accessibility of the web page. [Learn more about ARIA roles.](#)

Failing Elements

input.ng-tns-c65-11

input.ng-tns-c65-12

ARIA IDs are unique

The value of an ARIA ID must be unique to prevent other instances from being overlooked by assistive technologies. [Learn how to fix duplicate ARIA IDs.](#)

Failing Elements

input#drop-51

input#drop-51

input#drop-51

input#drop-51

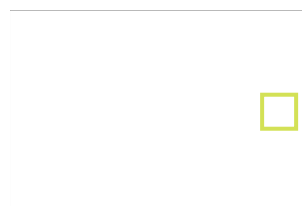
These are opportunities to improve the usage of ARIA in your application which may enhance the experience for users of assistive technology, like a screen reader.

NAMES AND LABELS

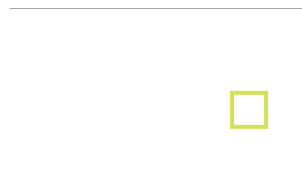
Image elements do not have `[alt]` attributes

Informative elements should aim for short, descriptive alternate text. Decorative elements can be ignored with an empty alt attribute. [Learn more about the alt attribute.](#)

Failing Elements

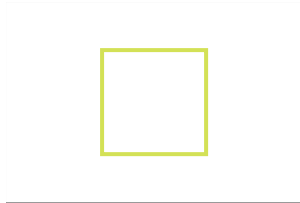


img

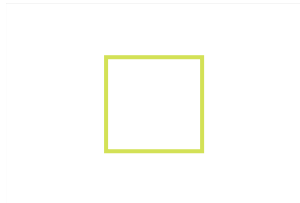


img

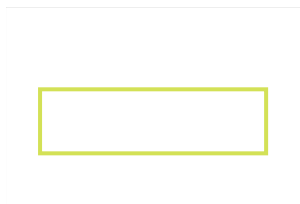
Failing Elements



img



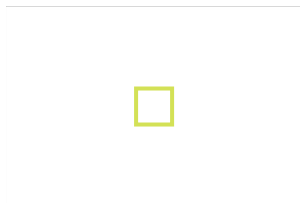
img



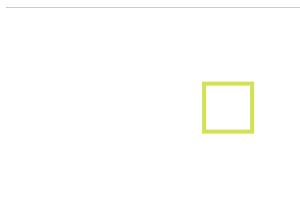
img



img



img



img#disha-banner-close

▲ `<frame>` or `<iframe>` elements do not have a title



Screen reader users rely on frame titles to describe the contents of frames. [Learn more about frame titles.](#)

Failing Elements



iframe

Failing Elements

iframe#chatbot

▲ Form elements do not have associated labels



Labels ensure that form controls are announced properly by assistive technologies, like screen readers. [Learn more about form element labels.](#)

Failing Elements

input.ng-tns-c57-8.ui-inputtext.ui-widget.ui-state-default.ui-corner-all.ui-autocomplete-input.ng-star-inserted

input.ng-tns-c57-9.ui-inputtext.ui-widget.ui-state-default.ui-corner-all.ui-autocomplete-input.ng-star-inserted

input.ng-tns-c58-10.ui-inputtext.ui-widget.ui-state-default.ui-corner-all.ng-star-inserted

▲ Links do not have a discernible name



Link text (and alternate text for images, when used as links) that is discernible, unique, and focusable improves the navigation experience for screen reader users. [Learn how to make links accessible.](#)

Failing Elements

a#dod

Failing Elements

These are opportunities to improve the semantics of the controls in your application. This may enhance the experience for users of assistive technology, like a screen reader.

CONTRAST

▲ Background and foreground colors do not have a sufficient contrast ratio.



Low-contrast text is difficult or impossible for many users to read. [Learn how to provide sufficient color contrast.](#)

Failing Elements



button.search_btn.train_Search



button.search_btn.train_Search



a



div.text-center.pull-right.card-new.scrollBtm.ng-star-inserted

a



body

These are opportunities to improve the legibility of your content.

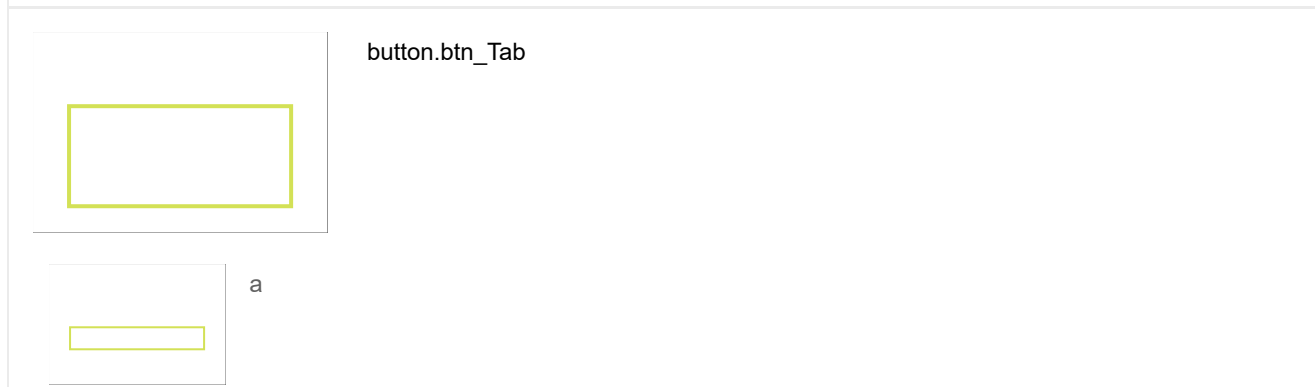
BEST PRACTICES

▲ Touch targets do not have sufficient size or spacing.



Touch targets with sufficient size and spacing help users who may have difficulty targeting small controls to activate the targets. [Learn more about touch targets.](#)

Failing Elements



These items highlight common accessibility best practices.

ADDITIONAL ITEMS TO MANUALLY CHECK (10)

Hide

☐ Interactive controls are keyboard focusable



Custom interactive controls are keyboard focusable and display a focus indicator. [Learn how to make custom controls focusable.](#)

☐ Interactive elements indicate their purpose and state



Interactive elements, such as links and buttons, should indicate their state and be distinguishable from non-interactive elements. [Learn how to decorate interactive elements with affordance hints.](#)

☐ The page has a logical tab order



Tabbing through the page follows the visual layout. Users cannot focus elements that are offscreen. [Learn more about logical tab ordering.](#)

☐ Visual order on the page follows DOM order



DOM order matches the visual order, improving navigation for assistive technology. [Learn more about DOM and visual ordering.](#)

☐ User focus is not accidentally trapped in a region



A user can tab into and out of any control or region without accidentally trapping their focus. [Learn how to avoid focus traps.](#)

- ☐ The user's focus is directed to new content added to the page ^

If new content, such as a dialog, is added to the page, the user's focus is directed to it. [Learn how to direct focus to new content.](#)

- ☐ HTML5 landmark elements are used to improve navigation ^

Landmark elements (<main>, <nav>, etc.) are used to improve the keyboard navigation of the page for assistive technology. [Learn more about landmark elements.](#)

- ☐ Offscreen content is hidden from assistive technology ^

Offscreen content is hidden with display: none or aria-hidden=true. [Learn how to properly hide offscreen content.](#)

- ☐ Custom controls have associated labels ^

Custom interactive controls have associated labels, provided by aria-label or aria-labelledby. [Learn more about custom controls and labels.](#)

- ☐ Custom controls have ARIA roles ^

Custom interactive controls have appropriate ARIA roles. [Learn how to add roles to custom controls.](#)

These items address areas which an automated testing tool cannot cover. Learn more in our guide on [conducting an accessibility review.](#)

PASSED AUDITS (18)

[Hide](#)

[aria-hidden="true"] is not present on the document <body> ^

Assistive technologies, like screen readers, work inconsistently when aria-hidden="true" is set on the document <body>. [Learn how aria-hidden affects the document body.](#)

[role]s have all required [aria-*] attributes ^

Some ARIA roles have required attributes that describe the state of the element to screen readers. [Learn more about roles and required attributes.](#)

Elements with an ARIA [role] that require children to contain a specific [role] have all required children. ^

Some ARIA parent roles must contain specific child roles to perform their intended accessibility functions. [Learn more about roles and required children elements.](#)

[aria-*] attributes are valid and not misspelled ^

Assistive technologies, like screen readers, can't interpret ARIA attributes with invalid names. [Learn more about valid ARIA attributes.](#)

Buttons have an accessible name

When a button doesn't have an accessible name, screen readers announce it as "button", making it unusable for users who rely on screen readers. [Learn how to make buttons more accessible.](#)

`[user-scalable="no"]` is not used in the `<meta name="viewport">` element and the `[maximum-scale]` attribute is not less than 5.

Disabling zooming is problematic for users with low vision who rely on screen magnification to properly see the contents of a web page. [Learn more about the viewport meta tag.](#)

ARIA attributes are used as specified for the element's role

Some ARIA attributes are only allowed on an element under certain conditions. [Learn more about conditional ARIA attributes.](#)

`[role]` values are valid

ARIA roles must have valid values in order to perform their intended accessibility functions. [Learn more about valid ARIA roles.](#)

Document has a `<title>` element

The title gives screen reader users an overview of the page, and search engine users rely on it heavily to determine if a page is relevant to their search. [Learn more about document titles.](#)

`<html>` element has a `[lang]` attribute

If a page doesn't specify a `lang` attribute, a screen reader assumes that the page is in the default language that the user chose when setting up the screen reader. If the page isn't actually in the default language, then the screen reader might not announce the page's text correctly. [Learn more about the `lang` attribute.](#)

`<html>` element has a valid value for its `[lang]` attribute

Specifying a valid [BCP 47 language](#) helps screen readers announce text properly. [Learn how to use the `lang` attribute.](#)

Links are distinguishable without relying on color.

Low-contrast text is difficult or impossible for many users to read. Link text that is discernible improves the experience for users with low vision. [Learn how to make links distinguishable.](#)

Lists contain only <code></code> elements and script supporting elements (<code><script></code> and <code><template></code>).	^
Screen readers have a specific way of announcing lists. Ensuring proper list structure aids screen reader output. Learn more about proper list structure.	
List items (<code></code>) are contained within <code></code> , <code></code> or <code><menu></code> parent elements	^
Screen readers require list items (<code></code>) to be contained within a parent <code></code> , <code></code> or <code><menu></code> to be announced properly. Learn more about proper list structure.	
No element has a <code>[tabindex]</code> value greater than 0	^
A value greater than 0 implies an explicit navigation ordering. Although technically valid, this often creates frustrating experiences for users who rely on assistive technologies. Learn more about the tabindex attribute.	
Heading elements appear in a sequentially-descending order	^
Properly ordered headings that do not skip levels convey the semantic structure of the page, making it easier to navigate and understand when using assistive technologies. Learn more about heading order.	
Deprecated ARIA roles were not used	^
Deprecated ARIA roles may not be processed correctly by assistive technology. Learn more about deprecated ARIA roles.	
Image elements do not have <code>[alt]</code> attributes that are redundant text.	^
Informative elements should aim for short, descriptive alternative text. Alternative text that is exactly the same as the text adjacent to the link or image is potentially confusing for screen reader users, because the text will be read twice. Learn more about the alt attribute.	

NOT APPLICABLE (26)

Hide

<input type="radio"/> <code>[accesskey]</code> values are unique	^
Access keys let users quickly focus a part of the page. For proper navigation, each access key must be unique. Learn more about access keys.	
<input type="radio"/> Elements with <code>role="dialog"</code> or <code>role="alertdialog"</code> have accessible names.	^
ARIA dialog elements without accessible names may prevent screen readers users from discerning the purpose of these elements. Learn how to make ARIA dialog elements more accessible.	
<input type="radio"/> ARIA input fields have accessible names	^

When an input field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more about input field labels.](#)

☐ ARIA `meter` elements have accessible names

When a meter element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn how to name meter elements.](#)

☐ ARIA `progressbar` elements have accessible names

When a progressbar element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn how to label progressbar elements.](#)

☐ `[role]`s are contained by their required parent element

Some ARIA child roles must be contained by specific parent roles to properly perform their intended accessibility functions. [Learn more about ARIA roles and required parent element.](#)

☐ Elements with the `role=text` attribute do not have focusable descendents.

Adding `role=text` around a text node split by markup enables VoiceOver to treat it as one phrase, but the element's focusable descendents will not be announced. [Learn more about the `role=text` attribute.](#)

☐ ARIA toggle fields have accessible names

When a toggle field doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more about toggle fields.](#)

☐ ARIA `tooltip` elements have accessible names

When a tooltip element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn how to name tooltip elements.](#)

☐ ARIA `treeitem` elements have accessible names

When a `treeitem` element doesn't have an accessible name, screen readers announce it with a generic name, making it unusable for users who rely on screen readers. [Learn more about labeling `treeitem` elements.](#)

☐ The page contains a heading, skip link, or landmark region

Adding ways to bypass repetitive content lets keyboard users navigate the page more efficiently. [Learn more about bypass blocks.](#)

☐ `<dl>`'s contain only properly-ordered `<dt>` and `<dd>` groups, `<script>`, `<template>` or `<div>` elements.

When definition lists are not properly marked up, screen readers may produce confusing or inaccurate output. [Learn how to structure definition lists correctly.](#)

☐ Definition list items are wrapped in `<d1>` elements ^

Definition list items (`<dt>` and `<dd>`) must be wrapped in a parent `<d1>` element to ensure that screen readers can properly announce them. [Learn how to structure definition lists correctly.](#)

☐ No form fields have multiple labels ^

Form fields with multiple labels can be confusingly announced by assistive technologies like screen readers which use either the first, the last, or all of the labels. [Learn how to use form labels.](#)

☐ `<html>` element has an `[xml:lang]` attribute with the same base language as the `[lang]` attribute. ^

If the webpage does not specify a consistent language, then the screen reader might not announce the page's text correctly. [Learn more about the lang attribute.](#)

☐ Input buttons have discernible text. ^

Adding discernable and accessible text to input buttons may help screen reader users understand the purpose of the input button. [Learn more about input buttons.](#)

☐ `<input type="image">` elements have `[alt]` text ^

When an image is being used as an `<input>` button, providing alternative text can help screen reader users understand the purpose of the button. [Learn about input image alt text.](#)

☐ The document does not use `<meta http-equiv="refresh">` ^

Users do not expect a page to refresh automatically, and doing so will move focus back to the top of the page. This may create a frustrating or confusing experience. [Learn more about the refresh meta tag.](#)

☐ `<object>` elements have alternate text ^

Screen readers cannot translate non-text content. Adding alternate text to `<object>` elements helps screen readers convey meaning to users. [Learn more about alt text for object elements.](#)

☐ Select elements have associated label elements. ^

Form elements without effective labels can create frustrating experiences for screen reader users. [Learn more about the select element.](#)

☐ Skip links are focusable. ^

Including a skip link can help users skip to the main content to save time. [Learn more about skip links.](#)

- Tables have different content in the summary attribute and `<caption>`. ^

The summary attribute should describe the table structure, while `<caption>` should have the onscreen title. Accurate table mark-up helps users of screen readers. [Learn more about summary and caption.](#)

- Cells in a `<table>` element that use the `[headers]` attribute refer to table cells within the same table. ^

Screen readers have features to make navigating tables easier. Ensuring `<td>` cells using the `[headers]` attribute only refer to other cells in the same table may improve the experience for screen reader users. [Learn more about the headers attribute.](#)

- `<th>` elements and elements with `[role="columnheader"/"rowheader"]` have data cells they describe. ^

Screen readers have features to make navigating tables easier. Ensuring table headers always refer to some set of cells may improve the experience for screen reader users. [Learn more about table headers.](#)

- `[lang]` attributes have a valid value ^

Specifying a valid [BCP 47 language](#) on elements helps ensure that text is pronounced correctly by a screen reader. [Learn how to use the lang attribute.](#)

- `<video>` elements contain a `<track>` element with `[kind="captions"]` ^

When a video provides a caption it is easier for deaf and hearing impaired users to access its information. [Learn more about video captions.](#)

■ Captured at Jan 19, 2025,
11:09 PM GMT+5:30

■ Initial page load

■ Emulated Desktop with
Lighthouse 12.2.1

■ Custom throttling

■ Single page session

■ Using Chromium 131.0.0.0 with
devtools

Generated by **Lighthouse** 12.2.1 | [File an issue](#)

Evaluation Table

Evaluation Criteria	Description	Rating (1-5)	Remarks	Action Items
Accessibility	Compliance with ARIA and WCAG standards	3	Several issues with ARIA roles and labels	Conduct a full accessibility audit and fix ARIA attributes
Performance	Loading speed and responsiveness	4	Good overall, but some delays observed	Optimize database queries and reduce page weight
Usability	Ease of navigation and user experience	2	Navigation is confusing and inconsistent	Redesign navigation layout and conduct user testing
Content	Clarity and accuracy of information	3	Some content lacks context and clarity	Revise content with clear guidelines
Security	Protection against vulnerabilities	4	Secure, but improvements needed for advanced threats	Implement advanced threat detection and regular security scans
SEO	Search engine optimization practices	3	Limited meta tags and alt text	Optimize metadata and add descriptive alt attributes

Suggested Improvements

1. ARIA Attributes

Issues:

- Mismatched or invalid `[aria-*)` attributes.
- Focusable descendants in `[aria-hidden="true"]` elements.
- Prohibited or incompatible ARIA roles.

Improvements:

- **Validate ARIA Attributes:** Use tools like WAVE or Axe to ensure `[aria-*)` attributes align with their parent roles.
 - **Avoid Hidden Focusable Elements:** Ensure elements within `[aria-hidden="true"]` are not interactive. Use `tabindex="-1"` or remove focusable attributes.
 - **Reassign Roles Properly:** Map ARIA roles only to compatible elements and remove prohibited roles.
-

2. Missing or Invalid Labels

Issues:

- Form elements without associated labels.
- Buttons, links, and menu items without accessible names.

Improvements:

- **Add Labels to Forms:** Ensure all input elements have `<label>` elements linked with `for` attributes.
 - **Accessible Button Names:** Use `aria-label` or `aria-labelledby` to provide descriptive names for buttons and links.
 - **Descriptive Link Text:** Avoid generic link text like "click here" and provide meaningful descriptions.
-

3. Image Alt Attributes

Issues:

- Images lack `alt` attributes or have redundant `alt` text.

Improvements:

- **Add Alt Text:** Provide concise, descriptive `alt` attributes for images conveying information.
 - **Decorative Images:** Use `alt=""` for images that are purely decorative to avoid redundancy for screen readers.
-

4. Contrast Ratio

Issues:

- Insufficient contrast between text and background.

Improvements:

- **Improve Contrast:** Adjust colors to meet WCAG 2.1 AA standards (minimum contrast ratio of 4.5:1 for normal text and 3:1 for large text).
 - **Contrast Testing:** Use tools like Contrast Checker or Lighthouse to verify updates.
-

5. Frame Titles

Issues:

- `<iframe>` elements lack descriptive titles.

Improvements:

- **Add Titles to Frames:** Provide meaningful `title` attributes to describe the content of `<iframe>` elements for screen readers.
-

6. Touch Target Size

Issues:

- Small or closely spaced interactive elements.

Improvements:

- **Increase Touch Targets:** Ensure buttons and links have a minimum target size of 48x48 pixels with adequate spacing.

7. Keyboard Navigation

Issues:

- Missing focus indicators on interactive elements.
- Inconsistent tab order.

Improvements:

- **Add Focus Styles:** Use `:focus` pseudo-class to make focus indicators visible for buttons, links, and form fields.
- **Ensure Logical Tab Order:** Verify and correct tabbing through elements to match the visual layout.

8. HTML Landmarks and Language

Issues:

- Lack of HTML5 landmark elements (e.g., `<main>`, `<nav>`).
- Missing or invalid `lang` attributes.

Improvements:

- **Add Landmarks:** Use `<header>`, `<main>`, `<footer>`, and `<nav>` to improve navigation for screen readers.
- **Specify Language:** Ensure the `<html>` element has a valid `lang` attribute (e.g., `lang="en"` for English).

9. Video Captions

Issues:

- `<video>` elements lack captions or transcripts.

Improvements:

- **Add Captions:** Include `<track>` elements with `kind="captions"` for all videos to provide accessibility for deaf or hearing-impaired users.