

# Fixing Accessibility Issues on Your Website: A Step-by-Step Guide

In today's digital age, ensuring your website is accessible to all users, including those with disabilities, is not just a legal requirement but also a moral imperative. Accessibility enhances user experience, broadens your audience reach, and ensures compliance with regulations like the Americans with Disabilities Act (ADA).

Drawing from my extensive experience on accessibility, I am sharing my insights and findings to help you tackle this important aspect of web development. This article will guide you through the process of identifying and fixing accessibility issues on your site using some of the best tools available.

## Tools to Use

1. **WAVE Evaluation Tool**
  - [Download WAVE](#)
2. **axe DevTools - Web Accessibility Testing**
  - [Download axe DevTools](#)
3. **Siteimprove Accessibility Checker**
  - [Download Siteimprove Accessibility Checker](#)
4. **NVDA SCREEN READER**
  - [Download NVDA SCREEN READER](#)

## Step-by-Step Guide

### 1. Install the Tools

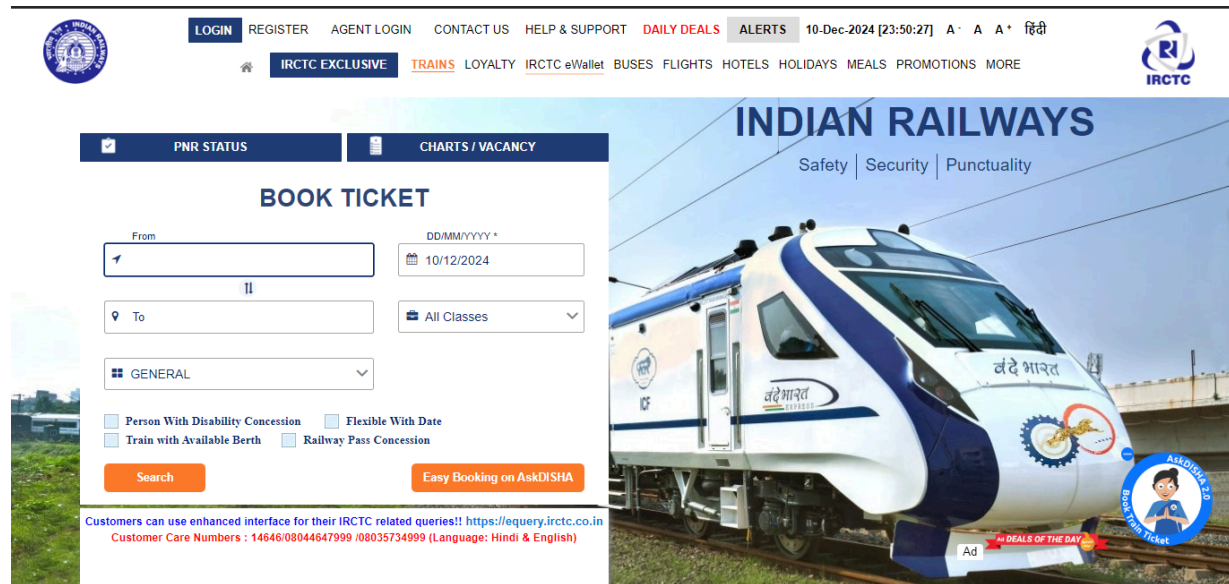
Begin by installing the above-mentioned tools on your web browser. Each tool offers unique features that will help you identify and fix different aspects of web accessibility.

### 2. Run Accessibility Tests and Fix Them

- **WAVE Evaluation Tool:** Start by scanning your website with the WAVE Evaluation Tool. This tool provides a comprehensive visual representation of your website's accessibility. It highlights errors, alerts, and features that need your attention, such as missing alternative text for images, color contrast issues, and structural elements that need improvement.
  - **Example:** Imagine you run an e-commerce site. You use WAVE to scan your product pages and discover that many product images lack alternative text. You also find that the color contrast between the text and background on your checkout page is insufficient. WAVE provides you with detailed information on these issues and suggests solutions.

- **How to Fix:** For instance, if the WAVE tool identifies missing alt text, add meaningful descriptions to images in the `alt` attribute. If it highlights low color contrast, you can use tools like the Color Contrast Analyzer to test and adjust the colors to meet WCAG guidelines.


E.g IRCTC Home Page



Issues reported by WAVE Extension:



Let's have a look at errors.



WAVE

web accessibility evaluation tool

powered by  
[WebAIM](#)

Styles: OFF ☐ ON ☒

Details

SummaryDetailsReferenceOrderStructureContrast

21 Errors

15 X Missing alternative text

5 X Missing form label


1 X Empty link

7 Contrast Errors

7 X Very low contrast

91 Alerts

2 X Redundant alternative text



LOGIN

\*aria-label="Click here to Login in application"\*

I.R.C.T.C."

\*tabindex=0\*

AGENT LOGIN

\*aria-label="

here to open contact us popup"

\*tabindex=0\*

HELP & S

DAILY DEALS

\*aria-label="Click here to open Daily De

2024 [23:53:04]

A

\*aria-label="decrease font size"\*

\*tabindex=0\*

हिंद

label="increase font size"

\*tabindex=0\*

हिंद

h3

h3

Home icon"

\*tabindex=

IRCTC EXCLUSIVE

\*aria-label="Me

TRAINS

\*aria-label="Menu Train"\*

L

\*aria-label="Menu LOYALTY"\*

IRCTC eWallet

\*aria-label="I.R.C.T.C. e.Wallet

BUSES

\*aria-label="Menu BUSES. Website w

Daily Deals has a contrast issue, we can get more info and fix it by clicking on the i icon next to the errors.



### 3. Identify and Prioritize Issues with axe DevTools

- **axe DevTools:** Use axe DevTools to perform a detailed accessibility audit. This tool integrates seamlessly into your browser's developer tools and provides comprehensive reports on identified issues. It categorizes issues into different severity levels: critical, serious, moderate, and minor. After installing the extension, it can be accessed from the DevTools section of your browser.
  - **Example:** While testing a blog page on your website, axe DevTools identifies several issues, including inaccessible form controls on your comment section and missing ARIA labels on navigation links. The tool categorizes these as serious issues and provides recommendations for fixing them.
  - **Technical Details:** axe DevTools can identify specific issues such as insufficient color contrast, missing ARIA attributes, inaccessible form controls, and focus management problems. The tool offers code snippets and guidance to fix each identified issue, making it easier for developers to implement the necessary changes.
  - **How to Fix:** When axe DevTools highlights an issue like a missing ARIA attribute, review the element in question and add the appropriate ARIA role or property. For example, if an interactive element is missing a role, you might add `role="button"` to make it recognizable to screen readers.

E.g IRCTC Home Page

The image shows a screenshot of the IRCTC Home Page with the axe DevTools extension open. The page displays the 'BOOK TICKET' form with fields for 'From', 'To', and 'Date'. The axe DevTools panel on the right shows a list of 24 issues, categorized by severity. The issues are as follows:

Issue Description	Count
Elements must only use supported ARIA attributes	3
ARIA commands must have an accessible name	2
Elements must only use permitted ARIA attributes	3
ARIA attributes must conform to valid values	2
Elements must meet minimum color contrast ratio thresholds	5
Images must have alternative text	5
Form elements must have labels	3
Interactive controls must not be nested	1


There are 24 issues reported by axe, brief description and fix to be done can be found in each issue's detail.


### ARIA commands must have an accessible name

2 ▾

Highlight Share Issue 1 of 2 << < > >>

Ensure every ARIA button, link and menuitem has an accessible name

[more information](#) 

Element Location: `.ui-dropdown-trigger.ui-corner-right[role="button"]` 

```
<div role="button" aria-haspopup="listbox" class="ui-dropdown-trigger ui-state-default ui-corner-right ng-tns-c65-11"><span class="ui-dropdown-trigger-icon ui-clickable ng-tns-c65-11 pi pi-chevron-down"></span></div>
```

To solve this problem, you need to fix at least (1) of the following:

- Element does not have text that is visible to screen readers
- `aria-label` attribute does not exist or is empty
- `aria-labelledby` attribute does not exist, references elements that do not exist or references elements that are empty
- Element has no title attribute

#### 4. Validate Content with NVDA Screen Reader

- **NVDA Screen Reader:** Test your website's accessibility by navigating it with NVDA Screen Reader. This will help you understand how users relying on screen readers interact with your site. Ensure that all interactive elements, such as buttons, links, and forms, are properly labeled and provide meaningful feedback to screen reader users.
  - **Example:** As you navigate your website using NVDA, you notice that the screen reader does not announce the presence of certain buttons, making it difficult for users to understand their purpose. Additionally, dynamic content updates, such as notifications, are not being announced to the user.
  - **Technical Details:** Ensure that ARIA roles and properties are correctly implemented, and that dynamic content updates (e.g., using JavaScript) are communicated to screen readers. Test the tab order of interactive elements to ensure logical navigation, and make sure all form elements have associated labels.
  - **How to Fix:** If you find that form elements are not correctly labeled, use the `aria-labelledby` or `aria-label` attributes to provide the necessary labels.

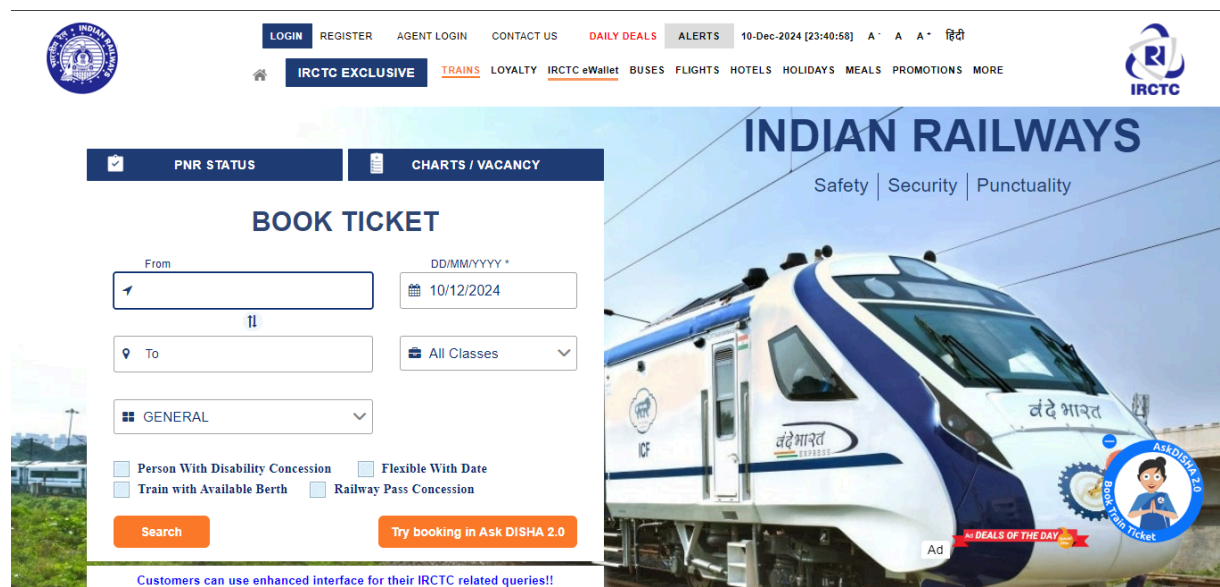


Ensure that dynamic content changes are announced to screen readers by using ARIA live regions, e.g., `<div aria-live="polite">`.

## 5. Continual Monitoring with Siteimprove

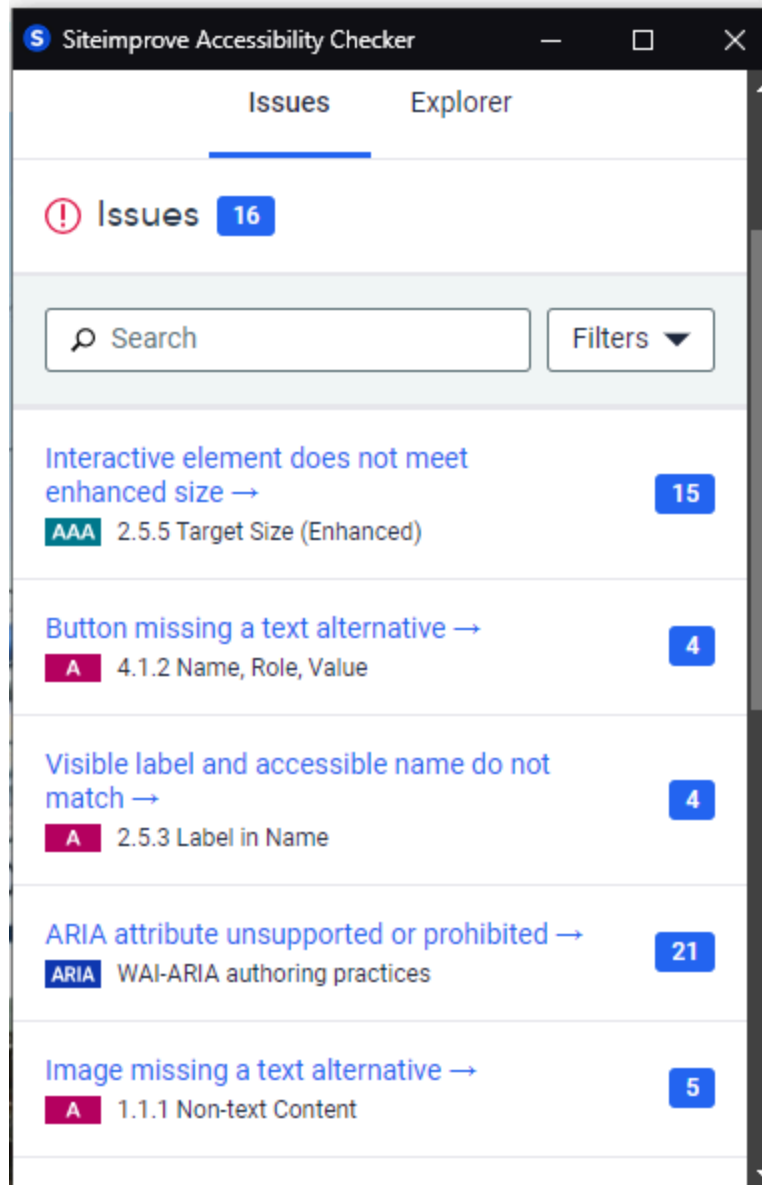
- **Siteimprove Accessibility Checker:** Use Siteimprove for continuous monitoring of your website's accessibility. The tool provides detailed insights into your site's accessibility status and tracks improvements over time.
  - **Example:** You have recently revamped your website and want to ensure it remains accessible. Siteimprove helps you monitor ongoing changes and provides detailed reports on areas needing attention, such as missing link descriptions and improper use of HTML5 elements.
  - **Technical Details:** Siteimprove offers a robust dashboard with metrics and benchmarks for different accessibility standards (e.g., WCAG 2.1). It can identify issues such as missing link descriptions, improper use of HTML5 elements, and areas where keyboard accessibility is lacking.
  - **How to Fix:** To address issues highlighted by Siteimprove, follow their detailed suggestions. For example, if Siteimprove finds missing link descriptions, add descriptive text inside the `<a>` tags that clearly indicates the link's purpose.

E.g IRCTC Website



The screenshot displays the IRCTC website's booking interface. At the top, there's a navigation bar with links like LOGIN, REGISTER, AGENT LOGIN, CONTACT US, DAILY DEALS, and ALERTS. Below this is a secondary bar with IRCTC EXCLUSIVE, TRAINS, LOYALTY, IRCTC eWallet, BUSES, FLIGHTS, HOTELS, HOLIDAYS, MEALS, PROMOTIONS, and MORE. The main content area features a 'BOOK TICKET' form with fields for 'From', 'To', 'Date' (DD/MM/YYYY), and 'Classes'. There are also checkboxes for various concessions like 'Person With Disability Concession', 'Flexible With Date', 'Train with Available Berth', and 'Railway Pass Concession'. A 'Search' button is present. To the right of the form is a large image of a train with the text 'INDIAN RAILWAYS' and 'Safety | Security | Punctuality'. A small circular logo in the bottom right corner of the image says 'Book Train Ticket'.

Issues reported by Siteimprove extension for IRCTC homepage

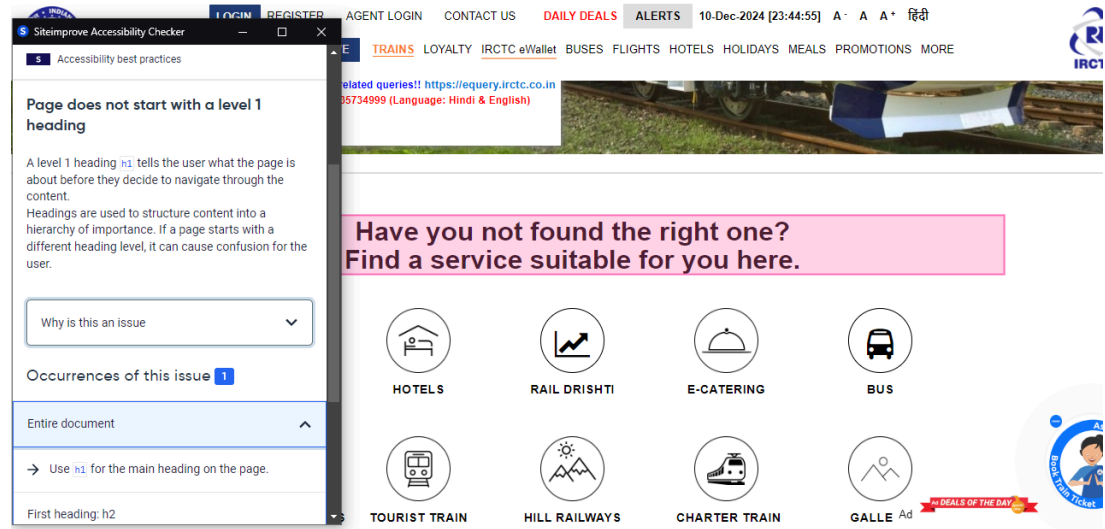


Resolution: All these issues can be resolved easily by going into each one of these issues and applying the fixes suggested by the tool.

**Note:** We need to fix issues with A and AAA labels, S labelled issues can be worked upon or ignored as they are best practices and not strictly required.

For the below issue which is a best practice, heading can be fixed by using h1 instead of h2, but it's totally as per our requirement how we want our site to look.





## Tips for Fixing Common Accessibility Issues

- **Use Semantic HTML:** Ensure your HTML elements are used correctly. For example, use `<header>`, `<nav>`, `<main>`, and `<footer>` tags to define the structure of your webpage. This helps screen readers understand the layout and structure of your content.
  - **Technical Details:** Use elements like `<article>`, `<section>`, and `<aside>` to semantically organize content. For example, group related content within a `<section>` tag and provide a heading.
  - **Example:** On a news website, use `<article>` for individual news stories and `<section>` to group related stories together.
- **Add Alt Text to Images:** Provide descriptive alt text for all images to help screen readers convey the content to visually impaired users. Ensure the alt text is concise yet descriptive enough to convey the image's purpose.
  - **Technical Details:** For decorative images, use `alt=""` to prevent screen readers from announcing them unnecessarily. For informative images, describe the content and purpose, e.g., `alt="Map showing the location of our office"`.
  - **Example:** For an e-commerce product page, an image of a red dress might have the alt text: `alt="Women's red dress, knee-length with a floral pattern"`.
- **Ensure Keyboard Navigation:** Make sure all interactive elements are accessible via keyboard navigation. Test using the `Tab` key to navigate through your site and ensure all focusable elements receive focus in a logical order.
  - **Technical Details:** Use the `tabindex` attribute wisely to control the tab order of elements. Ensure that custom components and interactive elements are keyboard accessible.

- **Example:** For a login form, ensure that users can tab through each field (username, password) and submit buttons in a logical sequence.
- **Color Contrast:** Check that there is sufficient color contrast between text and background to make the content readable for users with visual impairments. Use tools like the axe DevTools to automatically check and suggest color combinations.
  - **Technical Details:** Ensure a contrast ratio of at least 4.5:1 for normal text and 3:1 for large text. Use CSS to adjust colors, e.g., `color: #FFFFFF;`  
`background-color: #000000;`
  - **Example:** On a corporate website, ensure that the text on buttons has a high contrast against the button background color to be easily readable.
- **Responsive Design:** Ensure your website is fully responsive and functions well on all devices and screen sizes. Test your site on different devices and orientations to ensure consistent accessibility.
  - **Technical Details:** Use responsive design frameworks like Bootstrap or CSS media queries to adjust the layout for different screen sizes. Test using tools like Chrome DevTools' device mode.
  - **Example:** A restaurant website should adapt seamlessly to different devices, ensuring the menu is readable and buttons are accessible on both desktops and smartphones.
- **Focus Management:** Ensure that focus is managed correctly, especially for dynamic content. For instance, when a modal dialog opens, the focus should move to the first focusable element within the modal and should return to the previously focused element when the modal closes.
  - **Technical Details:** Use JavaScript to manage focus, e.g., `element.focus()`. Ensure the focus order is logical and intuitive.
  - **Example:** On a booking form, when a user selects a date, ensure the focus moves to the next logical field, such as the time selection.
- **Form Validation and Error Handling:** Make sure form validation messages are accessible. Use ARIA live regions to announce validation errors and make sure error messages are clear and actionable.
  - **Technical Details:** Implement client-side and server-side validation. Use `aria-invalid="true"` for invalid fields and `aria-describedby` to associate error messages with form fields.