



Touring - Touring BE

Accessibility audit

Client: Touring
Project: Touring BE
Date: 03/12/2024

Table of contents

| | |
|---|----------|
| Table of contents | 2 |
| Introduction | 4 |
| Scope of the evaluation | 4 |
| Out of scope | 4 |
| Sample pages for review | 4 |
| Summary | 5 |
| Priorities | 5 |
| 1. Keyboard | 5 |
| 2. Ajax | 5 |
| 3. Feedback & forms | 5 |
| 4. Visibility | 5 |
| 5. Content & other | 5 |
| Audit results | 6 |
| Success Criterion 1.1.1 Non-text content (minimum) (A) | 6 |
| Problem | 6 |
| Success criteria: | 6 |
| References: | 7 |
| Success Criterion 1.3.1 Info and Relationships (Minimum) (A) | 8 |
| Problem | 8 |
| Success criteria: | 8 |
| References: | 9 |
| Success Criterion 1.3.5 Identify Input Purpose (AA) | 10 |
| Problem | 10 |
| Success criteria: | 10 |
| References: | 10 |
| https://html.spec.whatwg.org/multipage/form-control-infrastructure.html#autofill-field | 10 |
| Success Criterion 1.4.3 Contrast (Minimum) (A) | 11 |
| Problem: | 11 |
| Success criteria: | 11 |
| References: | 12 |
| Success Criterion 1.4.13 Content on Hover or Focus | 13 |
| Problem: | 13 |
| Success criteria: | 13 |
| References: | 13 |
| Success Criterion 2.1.1 Keyboard (Minimum) (A) | 15 |

| | |
|---|-----------|
| Problem | 15 |
| Success criteria: | 16 |
| References: | 17 |
| Success Criterion 2.4.4 Link purpose (in context) (A) | 18 |
| Problem | 18 |
| Success criteria: | 18 |
| References: | 20 |
| Success Criterion 3.3.3 Error Suggestion (AA) | 21 |
| Problem | 21 |
| Success criteria: | 21 |
| References: | 22 |
| Success Criterion 2.4.7: Focus Visible (Minimum) (A) | 23 |
| Problem | 23 |
| Success criteria | 23 |
| References | 24 |
| Success Criterion 4.1.2 Name, Role, Value (Minimum) (A) | 25 |
| Problem | 25 |
| Success criteria: | 25 |
| References: | 25 |
| WAI ARIA 1.1 - ARIA for non-semantic HTML | 26 |
| Problem | 26 |
| Success criteria | 27 |
| References | 29 |
| Accessibility statement page | 30 |

Introduction

- This report describes the **conformance** of the submitted website with **W3C's Web Content Accessibility Guidelines (WCAG)**.
- A set of sample pages, for the submitted website or application, was tested and the WCAG Violations were documented with **code snippets and/or screenshots** when relevant.
- This report provides **advice** with **code suggestions** (not complete production code) and/or **links** to relevant resources
- The issues are **ordered by WCAG topic**
- We write **in English**, as most references and excerpts are from English-language WCAG documentation.
- To compose a **representative sample**, we base ourselves on a sampling of pages, aiming for at least 1 page per content type.
- Depending on the nature and complexity of the website, the **sample size** needed can vary, but we aim to include at least these:
 - The homepage,
 - A contact page (if it exists)
 - Representative forms
 - Basic page, eg. an 'about us'-page (or similar content page).
 - The aforementioned content types overview and a detail page

Scope of the evaluation

- **Website name:** Touring
- **Website base url:** touring.be
- **WCAG Version:** 2.2
- **Conformance target:** AA
- **Additional evaluation requirements:** *not provided*

Out of scope

The following are out of scope for this review:

- Integrated external services (eg. Google maps, embedded video players, ...), as these should have their own documentation about accessibility. If they are not accessible, this should be noted in the Accessibility Statement page or an alternative found.
- Logged-in sections (eg. admin area)
- Subsites or subdomains (eg. <https://jobs.touring.be/>)

Sample pages for review

See the audit results for links and context

Summary

Based on this evaluation, the website does not meet WCAG 2.2, Conformance Level AA. Detailed review results are available below.

Priorities

We recommend first focussing on elements of higher impact. These often center around keyboard navigation, Ajax and visibility. This list orders the priorities from higher to lower.

1. Keyboard

1. Success Criterion 2.1.1 Keyboard (Minimum) (A)
2. Success Criterion 2.4.7: Focus Visible (Minimum) (A)

2. Ajax

1. WAI ARIA 1.1 - ARIA for non-semantic HTML
2. Success Criterion 3.2.2 On Input (A)
3. Success Criterion 1.4.13 Content on Hover or Focus

3. Feedback & forms

1. Success Criterion 3.3.3 Error Suggestion (AA)
2. Success Criterion 4.1.2 Name, Role, Value (Minimum) (A): labels for input fields

4. Visibility

1. Success Criterion 1.4.3 Contrast (Minimum) (A)

5. Content & other

1. Success Criterion 1.1.1 Non-text content (minimum) (A)
2. Success Criterion 2.4.4 Link purpose (in context) (A)
3. Success Criterion 1.3.1 Info and Relationships (Minimum) (A)
4. Success Criterion 1.3.5 Identify Input Purpose (AA)

Audit results

Success Criterion 1.1.1 Non-text content (minimum) (A)

All non-text content that is presented to the user has a text alternative that serves the equivalent purpose, except for the situations listed below.

Problem

- **General:**
 - **Images:** not all images have an alt-attribute. Even if decorative, there needs to be an alt-attribute printed, with no value (`alt=""`).
 - Eg. Homepage -> **Articles plus récents** -> 2nd, 3rd and 4th images have no alt
 - Eg. the [‘Magazine’](#) page -> image grid, some are missing an alt
 - Eg. [Itinéraires rando et vélo en Belgique](#) -> images missing alt
 - ...
 - **Meaningful icons:** there are tables with checkmark and ‘x’ icons to signify if something is applicable or not. This needs to be part of the text or it won't be read by screen readers
 - Eg. on the [‘Assurance’](#) pages
 - Eg. [on this page](#)
 - ...
- **Icon font:** most icons (eg. social media, in the footer) are added by FontAwesome as an icon font. This is done using a pseudo-element and the CSS property `content` to enter a character which is replaced by an icon font. This character is picked up by screen readers, even though it is decorative.
- **Logo:** the visual logo contains text but this is not represented in the alt text. This alt text is what will give meaning to the link (because there is no title), so it needs to be complete.
- **Homepage:**
 - **Ce que nos clients racontent:** the slides contain a star rating but this value (eg. ‘5 stars’) is not communicated to screen-readers.
- [Jobs -> Postuler](#):
 - **Prêt à vous lancer:** this contains an infographic but there is no text alternative for it

Success criteria:

- **General:**
 - **Images:** change the image rendering template to **always** print an alt attribute, even when there is no description (then the alt would have an empty value: `alt=""`).
 - **Meaningful icons:** add some (visually hidden) text along with the icons

- **Icon font:** make sure the icon's CSS content is not picked up by screen readers. The best way to do this, is to use the icon styling on a separate HTML element, and put `aria-hidden=true` on it to hide it from screen readers. Eg. the 'stars' from the Carousel on the homepage has a wrapper which you can put that attribute on, or you can set it on the individual stars.
There is also [this alternative solution](#) purely done in CSS, but we can't guarantee it is future-proof. So the recommendation is using the separate HTML element. Or use the hack for a quick fix now, and do the replacement later.
- **Logo:** make the alt text for the logo 'Touring home'.
- **Homepage:**
 - **Ce que nos clients racontent:** add a visually hidden textual representation of the rating (eg. '5 stars'). If there is a page in the site where people interact with the rating, then you'll need to make markup changes instead. [See this example](#). If not, the visually hidden text should be enough.
- [Jobs -> Postuler:](#)
 - **Prêt à vous lancer:** the content of the image should be represented by pure text as well, since screen-readers can't read the image. This can be a visually hidden text block that simply contains a list with the 10 items, as well as the text and heading from above those items.
In contrast, on ['a propos de Touring'](#), the timeline visual is okay, because the content exists in the text.

References:

- **Source:** <https://www.w3.org/TR/WCAG21/#non-text-content>
- **Understanding:** <https://www.w3.org/WAI/WCAG21/Understanding/non-text-content.html>
- **Techniques:** <https://www.w3.org/WAI/WCAG21/quickref/#non-text-content>

Success Criterion 1.3.1 Info and Relationships (Minimum) (A)

Instructions provided for understanding and operating content do not rely solely on sensory characteristics of components such as shape, color, size, visual location, orientation, or sound.

Problem

- **General**
 - **Landmarks:** there are 6 or 7 `nav` tags in the page, these are normally useful as landmarks for screen readers to identify navigations, but with this many items they become a cluttered list.
 - **Lists:** text that visually functions as a list but doesn't use the list markup, should be fixed. They are often using checkmarks or 'x' icons:
 - [Occasions](#) (see the banner text)
 - [Dépannage](#) (2 lists)
 - <https://www.touring.be/fr/info-assurance-annulation>
 - [Assurance voyage](#) (multiple lists)
 - [Doctor Line](#) (at the bottom of the content)
 - ...
 - **Headings:** text styled to look like a heading, should be a heading
 - Eg. the 'INFORMATIONS LÉGALES' text found at the bottom of multiple pages [like this one](#) and other 'Assurance' pages.
- **Homepage:**
 - Structure of the headings needs to be logical and hierarchical. I'm seeing H2's that should be a lower level heading.
 - **News articles:** the heading should appear before the timestamp in the source code. This is the element that gives context to the teaser and is first to be read by a screen reader. Makes it confusing if a time stamp is read before the actual title.
 - Same issue on [Itinéraires rando et vélo en Belgique](#) (and other places with similar teasers)
- **'Magazine':**
 - Structure of the headings needs to be logical and hierarchical. I'm seeing H2's that should be a lower level heading.
- **jobs:**
 - There is inconsistency in the headings. The nested blocks in 'Nos départements' use H4 but the ones in 'Touring à la Une !' use H3.

Success criteria:

- **General**
 - **Landmarks:** you can replace some of the `nav` tags with `div`s. They will work just fine as just a list of links. Also make sure to remove the `aria-label` or `aria-labelledby` attributes

(they are not needed and not allowed on divs).

I suggest replacing the nav tags for these:

- The navigations in the footer: 'Au sujet de Touring' and 'Espace client'
- The navigation of the partners (icons)
- **Lists:** use unordered list markup (and)
- **Headings:** use headings in the markup: eg. H2, H3, ...
- **Homepage:**
 - Structure of headings: you should check any page with a teaser overview, for similar issues.
 - H2: Articles les plus récents
 - The teasers below that, should have H3 as a heading
 - **News articles:** move the heading before the timestamp in the source code, reposition using CSS.
 - Same issue on [Itinéraires rando et vélo en Belgique](#) (and other places with similar teasers)
- **'Magazine' :**
 - Structure of headings: you should check any page with a teaser overview, for similar issues.
 - H2: Catégories
 - The articles below that, should have H3 as a heading (including the Newsletter block)
- **Jobs:**
 - The various sections should use the same headings structure: H2 for the section title and H3 for the nested teasers or blocks, when they have them.

References:

- **Source:** <https://www.w3.org/TR/WCAG21/#info-and-relationships>
- **Understanding:** <https://www.w3.org/WAI/WCAG21/Understanding/info-and-relationships.html>
- **Techniques:** <https://www.w3.org/WAI/WCAG21/quickref/#info-and-relationships>

Success Criterion 1.3.5 Identify Input Purpose (AA)

Some people with cognitive disabilities may not understand the input's purpose from the label alone, so it's important to it easier to fill in forms. We can use code to indicate the purpose of common inputs.

Problem

- **Homepage:**
 - there is a newsletter subscription form with an email field that needs to be allowed to autofill
- **Other pages with forms:** similarly, any form that needs data from the user (an email, name, address, ...) should allow for autofill.

Success criteria:

- **Homepage:**
 - add this attribute to the email input field: `autocomplete="email"`
- **Other pages:**
 - Any form that needs data from the user (an email, name, address, ...), should be given an autocomplete property with the proper value ([see here for reference](#))
This is not needed for search forms, as there is no personal info to submit.

References:

- **Source:** <https://www.w3.org/TR/WCAG22/#identify-input-purpose>
- **Understanding:** <https://www.w3.org/WAI/WCAG22/Understanding/identify-input-purpose.html>
- **Techniques:**
 - <https://www.w3.org/WAI/WCAG22/Techniques/html/H98.html>

<https://html.spec.whatwg.org/multipage/form-control-infrastructure.html#autofill-field>

Success Criterion 1.4.3 Contrast (Minimum) (A)

The visual presentation of text, and images of text, needs to have a contrast ratio of at least **4.5:1**, **except** for the following:

- **Large Text:** text that is at least **24px** and images of large-scale text have a contrast ratio of at least **3:1**
- **Bold text:** at least **18.5 pixels**, the contrast is at least **3:1**
- **UI components** need a contrast of at least **3:1** (eg. think of form field borders)
 - User interface components are elements such as sub-menus, edit buttons, tabs, or thumbnails of media.
- **Incidental:** Text or images of text that are part of an inactive user interface component, that are pure decoration, that are not visible to anyone, or that are part of a picture that contains significant other visual content, have no contrast requirement.
- **Logotypes:** Text that is part of a logo or brand name has no contrast requirement.

Problem:

- **Homepage**
 - **Ce que nos clients racontent:**
 - this carousel uses a star rating as visual. Unfortunately, the contrast is too low.
 - The timestamps are too low contrast because of the light blue background
 - Similar thing in other pages with ratings, eg. [Assurance voyage](#)
- [June Energy](#)
 - White link on lightblue background has not enough contrast. See the text *“Rendez-vous sur votre compte MyTouring, et effectuez vos changements en quelques clics”*
- [Carrière:](#)
 - White text on grey banners, on a dark blue background: not enough text color contrast

Success criteria:

Ensure all these elements have sufficient color contrast between the text in the foreground and background color behind it.

Elements to fix:

- **Homepage**
 - **Ce que nos clients racontent:**
 - the contrast ratio for UI elements needs to be 3:1. The yellow is too light to go on a light color like that background. Either make the stars darker (**#968D03**) or put them on a darker background (**#3483E5**).
 - The timestamp needs to be darker to have a 4.5:1 contrast ratio (**#6B6B6B**)
 - Similar thing in other pages with ratings, eg. [Assurance voyage](#)
- [June Energy](#)
 - Change the text color to be dark, it needs at least 4.5:1 text contrast
- [Carrière:](#)

- White text on grey banners: make the text dark, it needs at least 4.5:1 text contrast

Contrast can be checked [using a color contrast checker](#) or the contrast checker in Chrome devtools.

References:

- **Source:** <https://www.w3.org/TR/WCAG22/#contrast-minimum>
- **Understanding:** <https://www.w3.org/WAI/WCAG22/Understanding/contrast-minimum.html>
- **Techniques:**
<https://www.w3.org/WAI/WCAG22/quickref/?showtechniques=143#contrast-minimum>

Success Criterion 1.4.13 Content on Hover or Focus

If hover or keyboard focus triggers additional content to become visible and then hidden, ensure interaction is **predictable**.

It also has to be:

- **Dismissable:** A mechanism is available to dismiss the additional content without moving pointer hover or keyboard focus, unless the additional content communicates an input error or does not obscure or replace other content;
- **Hoverable:** If pointer hover can trigger the additional content, then the pointer can be moved over the additional content without the additional content disappearing;
- **Persistent:** The additional content remains visible until the hover or focus trigger is removed, the user dismisses it, or its information is no longer valid.

Problem:

- [June Energy](#):
 - **Tooltips:** the fieldset 'Avez-vous votre numéro de client sous la main?' has a tooltip that is triggered by hover/focus. It is not picked up by screen-readers
 - See also other similar forms, eg. [Carteclient](#)

Success criteria:

- [June Energy](#):
 - **Tooltips:** there is little consensus on tooltips but there are a couple things to address here:
 - The content is loaded dynamically, which is not picked up by screen readers. You should at the very least have a container already present in the source code, in which the dynamic content will be displayed. You should also add `aria-atomic="true"` and `aria-live="polite"` on the tooltip content container so content change is communicated to the screen reader when it appears.
 - `role="tooltip"` has to be set on the content container, not on the triggering element. Also give it a unique ID.
 - `aria-describedby` attribute should be set on the trigger, and the value is the content container ID
 - Remove `aria-expanded` from the trigger (it is not allowed on span)

References:

- **Source:** <https://www.w3.org/TR/WCAG22/#content-on-hover-or-focus>

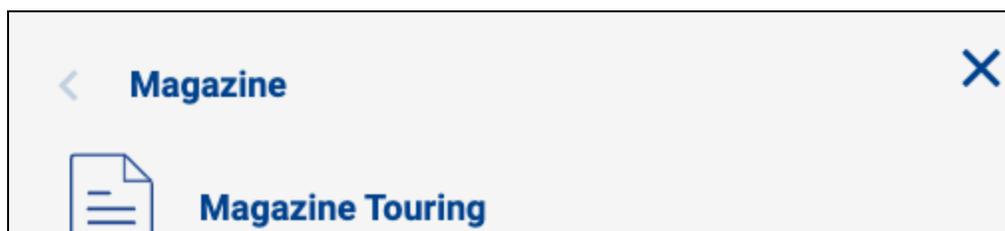
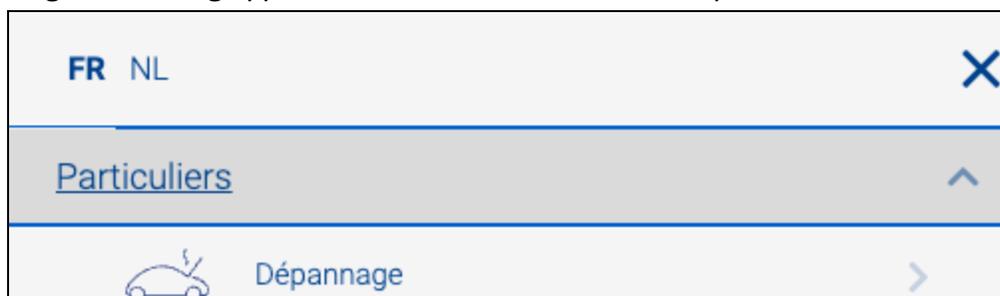
- **Understanding:**
<https://www.w3.org/WAI/WCAG22/Understanding/content-on-hover-or-focus.html>
- **Techniques:**
 - <https://www.w3.org/WAI/WCAG22/Understanding/content-on-hover-or-focus.html#techniques>
 - https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Roles/tooltip_role

Success Criterion 2.1.1 Keyboard (Minimum) (A)

All functionality of the content is operable through a keyboard interface without requiring specific timings for individual keystrokes, except where the underlying function requires input that depends on the path of the user's movement and not just the endpoints.

Problem

- **General:**
 - There is a link without a href: ``
- **Header:**
 - The **'language switcher'** can not be operated using the keyboard. You also can't focus on any of it
 - The **header search:** the collapsible field can not be operated using the keyboard but you can focus on the button. When activated, it sends you to a search page, which is not ideal.
 - The **main navigation:** when there are 'flyout' menus (submenus), this can not be accessed or operated by keyboard. Not the main items and not the flyout menus. Eg. See the homepage or the ['Magazine'](#) page.
 - **Mobile navigation:** you can't tab through the menu. When you get to 'Particuliers', a 'Magazine Touring' appears. You also can't close it or move past it.



- **Homepage:**
 - **Ce que nos clients racontent:** The carousel can not be operated using the keyboard + it doesn't have other accessibility features either
- [Mobilis:](#)
 - Map:
 - There is a popup on the 'Calcul d'itinéraire' tab, that appears on page load, asking you to log in. This one should receive focus, just like the cookie banner does.

- The form for the map ('Calcul d'itinéraire') can't be operated by keyboard. Some input fields work but you can't focus on all the buttons and options in there.
 - Same for what is displayed on the map itself ('Routes'), you can get to some elements (eg. some UI controls for the map but not all of them) but not the markers and info panels on the map (eg. when activating the 'événements' checkbox in the map controls).
- **Travel Wizard:** see [WAI ARIA 1.1](#)

Success criteria:

- **General:**
 - This "**main-content**" link functions as the target for the Skip link. It should not be a link itself. Make it a div or span instead.
- **Header:**
 - **language switcher:** You should be able to focus on the menu and open it using your keyboard, in order to then select the language and change/operate it in that way
 - The **header search:** there's 2 suggestions for a solution
 - You could use a '[disclosure](#)' pattern to make an accessible, collapsible block that can be operated by keyboard. So the visible 'search' button needs to be a 'toggle' button that opens the panel and reveals the real search button. The focus should jump into the field when it is open, hitting Enter (or clicking the real search button) then triggers the submit. Also make sure the input field and any buttons have a focus style.
 - Or you need to remove the search form and replace it with a link (a-tag) that leads to the search page, which can then be operated to perform a search
 - The **main navigation:**
 - The menu items need to be focusable and have focus styles. This goes for the flyout as well.
 - You'll need to use a '[disclosure](#)' pattern (eg. the one for menu or menubar) to operate the flyouts: buttons to operate the menus (instead of `i`-tags), correct aria attributes, etc.
 - **Mobile navigation:** you should be able to tab through all the main and sub-items and go back. Take care to make sure the search also works and has focus styles. And make sure there is a 'focus trap' so you can't accidentally tab out of the navigation while it is still open.
- **Homepage:**
 - **Ce que nos clients racontent:** You should be able to reach the next/previous buttons by tabbing. See also [the WAI ARIA section](#) to make it actually accessible.
- [Mobilis:](#)
 - Since this is an external integration, we recommend checking the documentation or contacting the company that made this map component, to ask about Accessibility

options. Basically, it needs to be operable by keyboard and the forms need to be decently built to be readable by screen-readers.

- Eg. the popup asking for login, on the 'Calcul d'itinéraire' tab, should get focus, have a focus trap and be keyboard accessible (just like the cookie banner)
- Eg. the inputs in the form of the map should have labels
- Eg. make sure the buttons are buttons that can be focussed on
- ...
- **Travel Wizard:** see [WAI ARIA 1.1](#)

References:

- **Source:** <https://www.w3.org/TR/WCAG21/#keyboard>
- **Understanding:** <https://www.w3.org/WAI/WCAG21/Understanding/keyboard.html>
- **Techniques:**
 - <https://www.w3.org/WAI/WCAG21/quickref/#keyboard>
 - <https://www.w3.org/WAI/ARIA/apg/patterns/menubar/>

Success Criterion 2.4.4 Link purpose (in context) (A)

The purpose of each link can be determined from the link text alone or from the link text together with its programmatically determined link context, except where the purpose of the link would be ambiguous to users in general.

Problem

- **Footer:**
 - Some of the **Social Media** icon-links in the footer, don't have a (descriptive) text (eg. Instagram).
- **Homepage:**
 - **Big banner image:** The image alt text does is not descriptive but there is text embedded in the image itself.
 - **#block-homepage4miniblock:** in these teasers, you have the same link around the image and the title. The link around the image also has a very generic visually hidden text that just says 'Image'.
In addition, both links point to the same thing, that's redundant.
 - **En tant que client de Touring, vous méritez plus:** the links for the teasers have no descriptive text, only some visually hidden text called 'Image'.
 - **Articles les plus récents:** similar issue as with the 4 mini blocks, there's multiple versions of essentially the same link.
 - See also [Itinéraires rando et vélo en Belgique](#) (and other places with similar teasers)
- [Contact](#)
 - **The 6 icon blocks:** Structurally, it doesn't make sense to get the link before the title. This violates [Criterion 1.3.1](#). There's also again the meaningless 'Image' text that is visually hidden, in the image link and no decent link description to be seen. The alt text on these images is not very descriptive.
- [Péages et vignettes:](#)
 - Yet another type of teaser construction: links around teaser, without a description but containing a heading.
- [Blog articles](#)
 - Again, too many links and issue with the image link description.
 - It's the same with [the teasers in sidebars of articles](#).

Success criteria:

- **Footer:**
 - **Social Media** icon-links in the footer: make the Instagram link complete by filling in the title attribute.
- **Homepage:**

- **Big banner image:** a lot of time, images in banners will be decorative, but since this one contains text, we would recommend putting that text in the image. Especially because the image-text does not exactly match the banner title text. The title mentions there is a 20% discount, but does not mention on what exactly. While the image is more clear and says the discount applies to 'dépannage en Belgique'. It also contains the asterisk that points to the small bit of text below the banner.
- **#block-homepage4miniblock:** best would be to make the entire card clickable like this:
 - Remove the link around the image
 - Remove the generic 'Image' text that is visually hidden, you don't need it because the image alt-attribute has decent descriptive text the link can fall back on.
 - Make the entire card clickable by using CSS on the title link. [There's a pseudo-content technique that can help](#)
- **En tant que client de Touring, vous méritez plus:**
 - The alt text of the image functions as the description for the image link, so you should remove the 'Image' text that is visually hidden. This serves no purpose and it is also too generic.
- **Articles les plus récents :**
 - Remove the link around the image (and on any 'read more' link if there is any)
 - Make the entire card clickable by using CSS on the title link. [There's a pseudo-content technique that can help](#)
 - There is another link in there that is in some cases hidden using 'display:none' on the wrapper. That's okay, it works. But seems like that should just not be in the source-code. Ideally, you would remove the code itself, if at all possible.
 - Same changes needed on [Itinéraires rando et vélo en Belgique](#) (and other places with similar teasers)
- **Contact**
 - **The 6 icon blocks:** the link should be removed from the image, and instead a link should be put on the titles. Like the previous examples, you can use the pseudo-content technique to make the entire card clickable. Again, also remove the 'Image' text that is visually hidden.
I would also suggest setting an empty value for the image alt-texts, since these images are decorative.
- **Péages et vignettes:**
 - We recommend removing the link around the entire item and making a linked-heading (like in some of the previously mentioned teasers) using the pseudo-content technique to keep the entire card clickable. Remove the visibly hidden 'Image' text. That would make it consistent with the other teaser recommendations.
Alternatively, you could just add a link description + remove the 'Image' text.
Or even, replace the 'Image' text with the heading text. But the first solution is

best as it makes the teasers consistent, which makes screen reader navigation better.

- [Blog articles](#)
 - Again, make clickable cards using the title link, remove other links

References:

- **Source:** <https://www.w3.org/TR/WCAG22/#link-purpose-in-context>
- **Understanding:** <https://www.w3.org/WAI/WCAG22/Understanding/link-purpose-in-context.html>
- **Techniques:** <https://www.w3.org/WAI/WCAG22/quickref/#link-purpose-in-context>

Success Criterion 3.3.3 Error Suggestion (AA)

If an input error is automatically detected and suggestions for correction are known, then the suggestions are provided to the user, unless it would jeopardize the security or purpose of the content.

Problem

- **Homepage:**
 - **'Ne manquez rien de Touring'**: the error message when trying to submit without filling in an email address, does not mention what the error is (eg. is the field empty? Is it not a valid email address? ...) and does not suggest how to correct the error. It's also in English instead of French.
- [June Energy](#):
 - Similar issues, but I think this one is Ajax based, so some extra caveats apply
- [Carteclient](#):
 - Very similar to June Energy, but this one doesn't use Ajax I think. Not seeing real error messages appear on submit, just browser validation.

Success criteria:

- **Homepage:**
 - **'Ne manquez rien de Touring'**: as it turns out, we can't rely on native browser implementation of error messages for form fields.
 - You should have an `aria-invalid` attribute set on the **input** field, as well as an ID.
 - An [error message](#) (div) should be present underneath fields (eg. when submitting without filling in all required fields). This div should have an attribute `aria-errormessage` to match the field ID.
 - For the **message content**: it should be in the language of the page. Other than that, customize the message to include what is wrong and how to correct it. Eg. 'Fill in a valid email address'
- [June Energy](#):
 - There are individual **error message divs**, so that's good. However, they are missing the aforementioned aria attributes and ID's for the inputs (see previous issue)
 - Same for the **input fields**, they should have an `aria-invalid` attribute set as well as an ID
 - Since the submit validation is based on **Ajax**, the error message divs have to already exist in the source-code, with `aria-atomic="true"` and `aria-live="polite"` properties on them. Only the content should be dynamic.
- [Carteclient](#):

- This one has no error message divs, similar as '**Ne manquez rien de Touring**'. Unless there are some Ajax-validated fields, then the Ajax remarks from June Energy apply as well.

References:

- **Source:**
- **Understanding:** <https://www.w3.org/WAI/WCAG21/Understanding/error-suggestion.html>
- **Techniques:**
 - <https://www.w3.org/WAI/WCAG22/quickref/?versions=2.1#error-suggestion>
 - <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Attributes/aria-error-message>
 - <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Attributes/aria-describedby>

Success Criterion 2.4.7: Focus Visible (Minimum) (A)

This is partly custom styling, BUT the theme CSS should be checked to see that 'focus' states are not simply disabled by default for the elements described (eg. the Layout Builder Blocks buttons). That way it can be set up better by default in Rocketship.

The purpose of this success criterion is to help a person know which element has the keyboard focus. Authors are responsible for providing at least one mode of operation where the focus is visible. In most cases there is only one mode of operation so this success criterion applies. The focus indicator must not be time limited, when the keyboard focus is shown it must remain.

Note that a keyboard focus indicator can take different forms.

Problem

- **Search:**
 - There are duplicate ID's on the search forms. Eg. there are 2 search form blocks in the header (1 for mobile, 1 for desktop) and they should have unique ID's on the input field.
- **Forms:**
 - Some input fields are missing a visible focus style:
 - in header search form
 - Ne manquez rien de Touring (newsletter subscription block)
 - Dropdown select on [Carteclient](#) -> 'Type de contrat'
 - Toggle switches need to follow an ARIA pattern to be readable by screen-readers and operable by keyboards.
 - Eg. [lune energy](#) has a 'oui/non' switch in its form
 - Eg. [Carteclient](#)
- **Travel Wizard:** see [WAI ARIA 1.1](#)

Success criteria

- **Forms:**
 - Input fields need a reliable focus style:
 - Header search: make the border dark blue on focus
 - Newsletter subscription block: the blue border gets lost on the blue background. Consider making it another contrasting color when viewed on a dark background. Be careful to keep the contrast ratio in 3:1 (eg. #FF4D4D)
 - Firstly, the toggle switch on [lune energy](#) needs to get a focus style so it is visible when focussed by keyboard, but it also needs to be reconstructed using an accessible pattern ([like this example](#)). Aria attributes need to be added to make it workable by screen readers.
- **Travel Wizard:** see [WAI ARIA 1.1](#)

References

- **Source:** <https://www.w3.org/TR/WCAG21/#focus-visible>
- **Understanding:** <https://www.w3.org/WAI/WCAG21/Understanding/focus-visible.html>
- **Techniques:**
 - <https://www.w3.org/WAI/WCAG21/Understanding/focus-visible.html#techniques>
 - <https://www.w3.org/WAI/WCAG21/Techniques/general/G195>

Success Criterion 4.1.2 Name, Role, Value (Minimum) (A)

For all user interface components (including but not limited to: form elements, links and components generated by scripts), the name and role can be programmatically determined; states, properties, and values that can be set by the user can be programmatically set; and notification of changes to these items is available to user agents, including assistive technologies.

Problem

- **Header:**
 - The **search** field's submit button has no label

Success criteria:

- **Header:**
 - The **search:** the submit button needs to be labelled. Easiest way is to add a title + an aria-label attribute with the value 'Search'.

References:

- **Source:** <https://www.w3.org/TR/WCAG21/#name-role-value>
- **Understanding:** <https://www.w3.org/WAI/WCAG21/Understanding/name-role-value.html>
- **Techniques:** <https://www.w3.org/WAI/WCAG21/quickref/#name-role-value>
 - <https://www.w3.org/WAI/WCAG21/Techniques/html/H91>
 - <https://www.w3.org/WAI/WCAG21/Techniques/failures/F68>

WAI ARIA 1.1 - ARIA for non-semantic HTML

WAI-ARIA is used to create an accessible Rich Internet Application. There are recommended usage patterns that should be followed.

Languages used to create rich and dynamic web sites, e.g., HTML, JavaScript, CSS, and SVG, do not natively include all the features required to make sites usable by people who use assistive technologies (AT) or who rely on keyboard navigation.

So when using **design patterns (like dropdown menus)**, please refer to [the ARIA documentation](#) to apply the correct attributes and make these patterns accessible but note that simply applying ARIA attributes does NOT automatically apply styling or behaviour. You still need JS and CSS for that. ARIA simply adds elements to the page's accessibility tree.

Problem

- **Navigation**
 - **Mobile:** the open/close actions are not communicated properly to screen-readers
- **Homepage:**
 - **Ce que nos clients racontent:** this **Carousel Section** doesn't handle focus and keyboard navigation very well. So it fails at least [Success Criterion 2.1.1 \(keyboard\)](#) but it is also missing features to make it usable by screen readers.
- [Jobs:](#)
 - The block with the collapsed items (FAQ): this component is not accessible, as you never enter the component or focus on anything while using keyboard or an assistive tool. Making it fail [Success Criterion 2.1.1 \(keyboard\)](#) and use by screen readers in general.
- [Contact](#)
 - Same thing with the FAQ in here
- [FAQ:](#)
 - Same for the **FAQ** in here (a similar one is also on [June Energy](#) and [Carteclient](#))
 - **Search field:** this one has an auto-suggest list when you type. But it's not communicated to a screen-reader that this appears. That violates the [Content on Focus criterium](#)
 - **Facets:** it's not clear by itself, what this list is or does.
- [Product simulators:](#)
 - There are interactive modules (eg. [this one for Ski insurance](#)), which can not be operated by keyboard and are also not fully screen-reader friendly
- [Travel wizard:](#) violates [Success Criterion 2.1.1 \(keyboard\)](#) and [2.4.7 Focus Visible](#)
 - The step-navigation needs to be communicated to screen-readers
 - The options don't have a focus style, making it impossible to see they are focussed on (eg. when using keyboard)
- **Forms with Ajax messaging or loaders**

- Near the oui/non toggle ('numéro du client'), there is interaction that triggers a preloading message + new fields appears. These type of dynamic content are not ready by screen-readers.
 - Eg. [Carteclient](#)
 - Eg. [lune Energy](#)

The image shows a user interface for a form. At the top, there is a blue header with the text "Je connais mon numéro de client *". Below this is a toggle control with two buttons: "Oui" (highlighted in blue) and "Non" (greyed out). Underneath the toggle is the label "Numéro client Touring" followed by a text input field. The input field contains the text "Numéro client Touring". At the bottom of the form is a yellow button with the text "Contrôle des données remplies".

Success criteria

- **Navigation**
 - **Mobile:** you need to use a disclosure pattern for this
 - the open & close links, should be buttons. They are action triggers, not navigation links.
 - Give the button text: 'Menu' for the open action, and 'Close' for the close action
 - Set `aria-expanded="false"` on the Menu button, and toggle the value when it is open or closed ('false' when closed, 'true' when open)
 - Set an ID on the header--navigation div, and set `aria-controls` attribute on the Menu button, which includes that ID as its value
- **Homepage:**
 - **Ce que nos clients racontent:** you can refer to [this pattern found in here](#) (and the included examples code). This use case might be a bit different, but the important part is that you integrate the **Basic carousel elements** (various aria-attributes) + keyboard navigation can be used to cycle through and all slides are reachable/readable.
The slides also have `tabindex=0` set to them, which is pretty useless because the slides are not interactive elements. It's generally not allowed to set `tabindex` on elements that don't have a role that supports focus.
- [Jobs:](#)

- **The block with the collapsed items (FAQ):** since this component is a form of a 'disclosure' component, take a look at [the disclosure WAI ARIA pattern](#) & [FAQ example](#) for inspiration on how to change the HTML (and possibly JS) to make it accessible and usable via keyboard (button, aria attributes, ...). Alternatively, you could use native html (summary and details elements). I would also recommend using H3's for the item titles, since they act as headings.
- [Contact](#)
 - Same thing with the FAQ in here
- [FAQ:](#)
 - **Search field:** it would not be handy to add another submit button just to open up the list of suggestions, so what you could do, is add a bit of text above the search form, to explain that there is an auto-suggestion option that appears when you type.
 - **Facets:** Add a heading H2 in the sidebar, above the facets (it can be visually hidden) and call it 'Filters' or 'Tags' (or whatever taxonomy name fits best), so screen-readers can pick up that this is a list of filters.
 - The **overview items block** needs 2 things:
 - an H2 (can be visually hidden) called 'Results', to name the section and form a parent for the item's H3's.
 - The collapsible list (FAQ) needs to be formatted properly to be accessible. See the previous examples listed in this section (eg. from Contacts and Jobs).
- [Product simulators:](#)
 - There are several components that need to change:
 - **Toggle switches:** you can find [patterns for these](#) to make them accessible
 - There are differently styled toggles in [June energy](#) and other forms, these have extra issues described below in '**Forms with Ajax messaging or loaders**'
 - **Number fields:** (increase/decrease controls) put the '+' and '-' buttons together in the source code, after the number input itself. Keyboard users need to be able to operate those in succession without having something else focussable inbetween them.
There should also be an ID on the input + `aria-controls` attribute on each button. The aria attribute then uses the input field ID as its value. That way, screen readers know what controls are tied to what fields.
 - **Calendar popout:** this needs to be reachable via keyboard (tabbing) and operable as well (change dates).
- [Travel wizard:](#)
 - For screen-readers to recognize the step navigation as a sort of breadcrumb, you'll need some aria attributes. Eg. [use a pattern like this one](#).
The list needs to be an **ordered** list and you'll need at least a `aria-label="breadcrumb"` and `aria-current`.
 - The focus is not visible when you use keyboard to reach the options. This focus needs to be visible for users to operate it.

- There's a number field in some steps, these should have the same changes as mentioned in the 'Product simulators' review (see item above)
- [The result page](#) needs a heading. Make 'Résultats' an H2 for example. Or if there is no page title yet, you can use an H1. This would be more consistent with the Steps of the Wizard (they also use H1 for their pages)
- **Forms with Ajax messaging or loaders**
 - Any preloader message or new text content that gets added because of interaction, needs to have a wrapper in the source code, with aria attributes added to make it detected by screen readers: `aria-atomic="true"` and `aria-live="polite"`. The content is then added (or shown) dynamically.
 - The toggle has multiple problems:
 - If a toggle makes new field appear, adding `aria-controls` and `aria-expands` is needed (like in this [disclosure pattern](#)).
 - But the toggle also needs to be accessible, there are [patterns for these](#)

References

- **Examples & suggestions:**
 - https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA_Live_Regions#basic_example_dropdown_box_updates_useful_onscreen_information
 - <https://connekthq.com/accessibility-and-ajax-load-more/#feature-aria-regions>
 - <https://dev.to/garybyrne/introduction-to-aria-live-regions-2mOg>
- **Source:** <https://www.w3.org/TR/wai-aria-1.1/>
- **Practices:** <https://www.w3.org/WAI/ARIA/apg/practices/>
- **Patterns:** <https://www.w3.org/WAI/ARIA/apg/patterns/> (eg. for disclosure/faq, ...)

Accessibility statement page

What is (and isn't) part of that page:

- (en) <https://www.w3.org/WAI/planning/statements/>
- (nl) <https://overheid.vlaanderen.be/toegankelijkheidsverklaring>

How to:

- <https://www.anysurfer.be/nl/documentatie/artikels/detail/toegankelijkheidsverklaring>
- <https://www.w3.org/WAI/planning/statements/>

Generate a page

- <https://www.w3.org/WAI/planning/statements/generator/#create>

Examples:

- (en) <https://usability.yale.edu/web-accessibility/accessibility-yale>
- (nl) <https://ibz.be/nl/toegankelijkheidsverklaring>