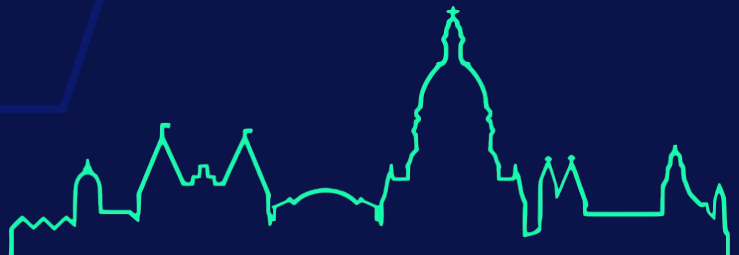
hyvä

Accessibility Testing with AI

Train your AI
to become a A11Y Expert



< TODAY'S SPEAKER />

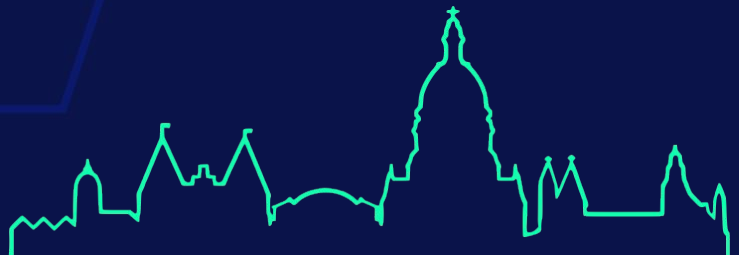
Sean van Zuidam

Frontend Wizard @ Hyvä

@grimlink.com GrimLink



$whoami



What is Good A11Y?

What is Good A11Y?

01. CONTRAST

Visual Contrast

Good contrast, *not just color*.

02. SEMANTICS

Semantic HTML

Semantic HTML, stop with div soup!

03. OPERABILITY

Keyboard Control

Operable with a Keyboard and other devices.

04. ANIMATION

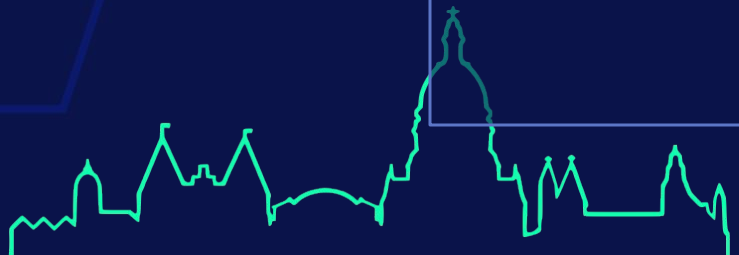
Motion Control

Too many animations.

05. LABELS

Clear Labeling

Use proper labels, *click here* is not clear.



Why AI assistants get A11Y wrong

01. HTML SEMANTICS

Misusing HTML Structure

Using HTML the wrong way, such as creating a <section> without an accessible label.

02. NATIVE ELEMENTS

Ignoring Native Controls

Using role=dialog on a generic instead of leveraging native dialog elements.

03. COMPLEXITY

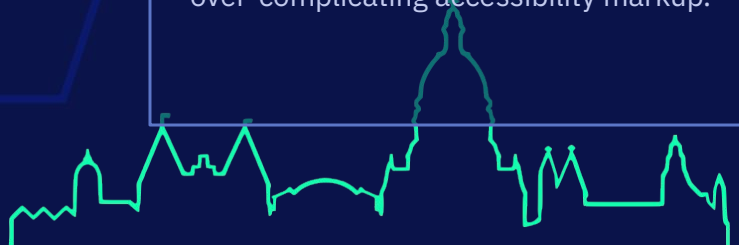
Over-Engineering A11Y

Adding excessive aria attributes and over-complicating accessibility markup.

04. INPUT HANDLING

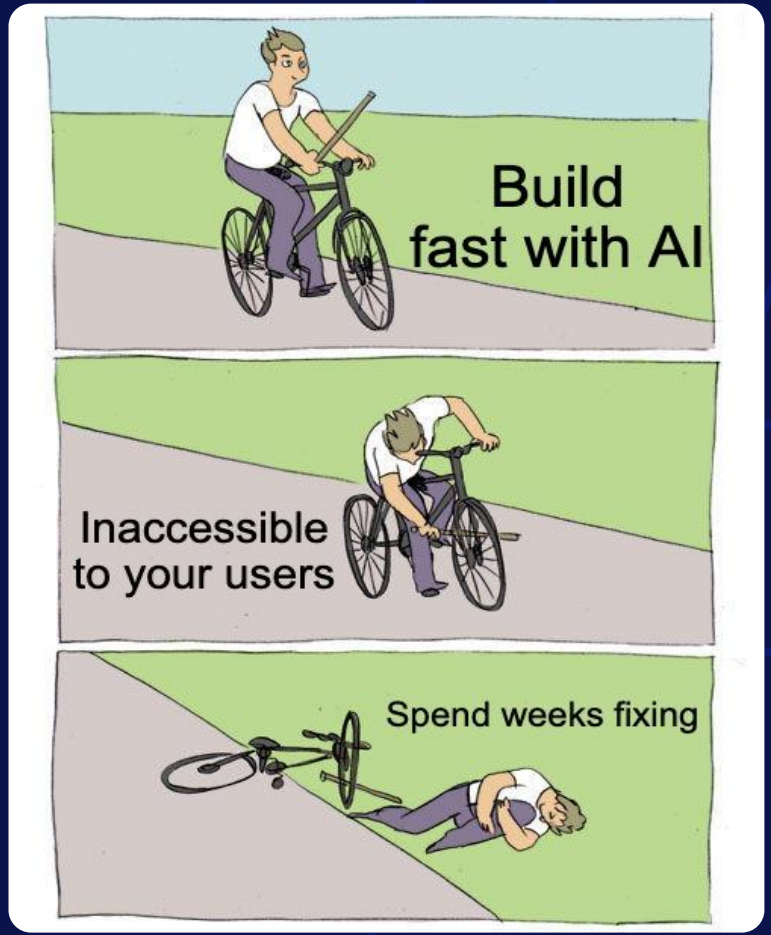
Redundant Attributes

Using text input with inputmode="number" and extra markup instead of a simple type="number".



Some Real Cases

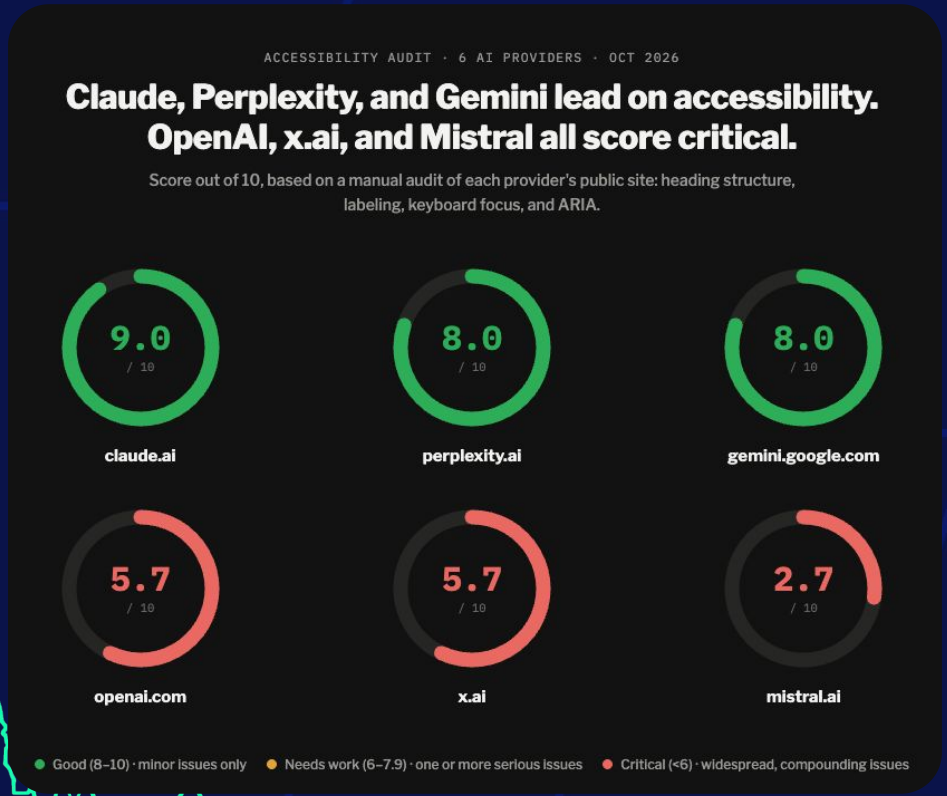
Let's look at how current AI Tools and this Event actually performs.



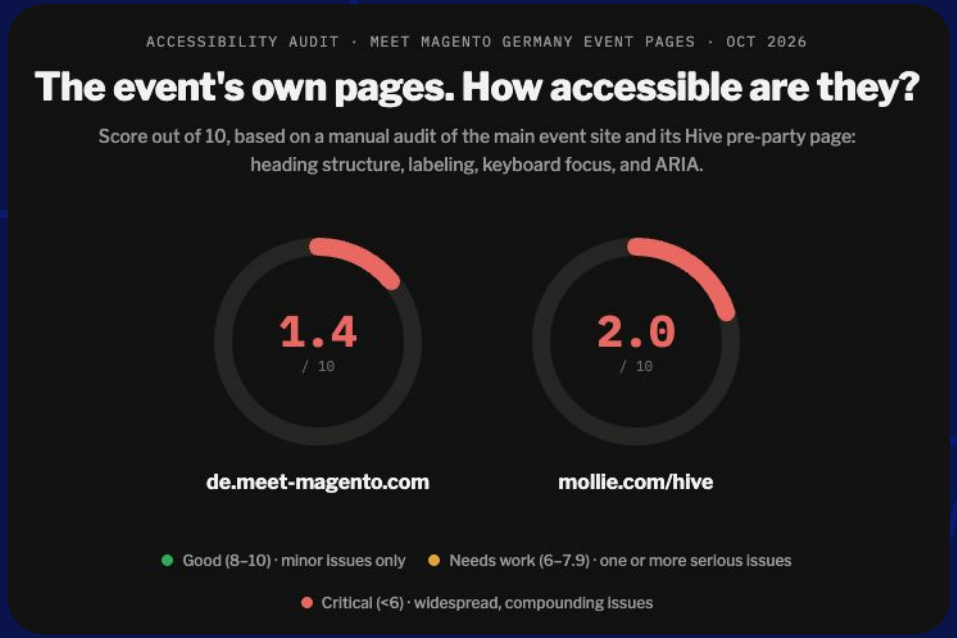
Small Disclaimer:

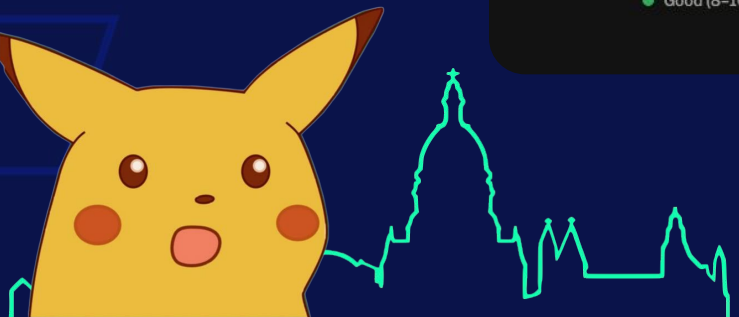
We are having fun not pointing at mistakes.

Are AI sites A11Y?



How about this event and Hive?





Few notes on the scoring

AUTOMATED TESTING

Lighthouse & Checkers

Automated tools check a page **once per rule** (e.g. "links have names": pass/fail), not once per instance.

A page can fail that rule a single time and still score well overall.

MANUAL AUDIT

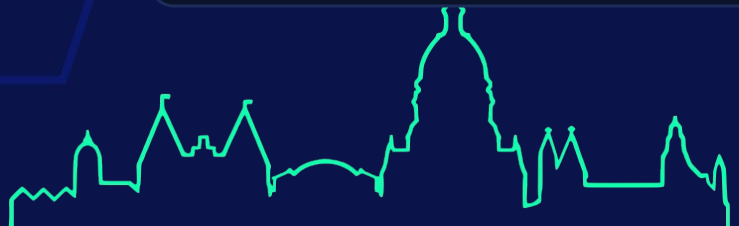
A11y Tree & DOM

Done manually per the page's actual accessibility tree, keyboard behavior, and DOM. Penalizes per distinct issue found:

Serious −2

Moderate −1

Minor −0.3



How about the Organizers?

ACCESSIBILITY AUDIT · MEET MAGENTO GERMANY ORGANIZERS · OCT 2026

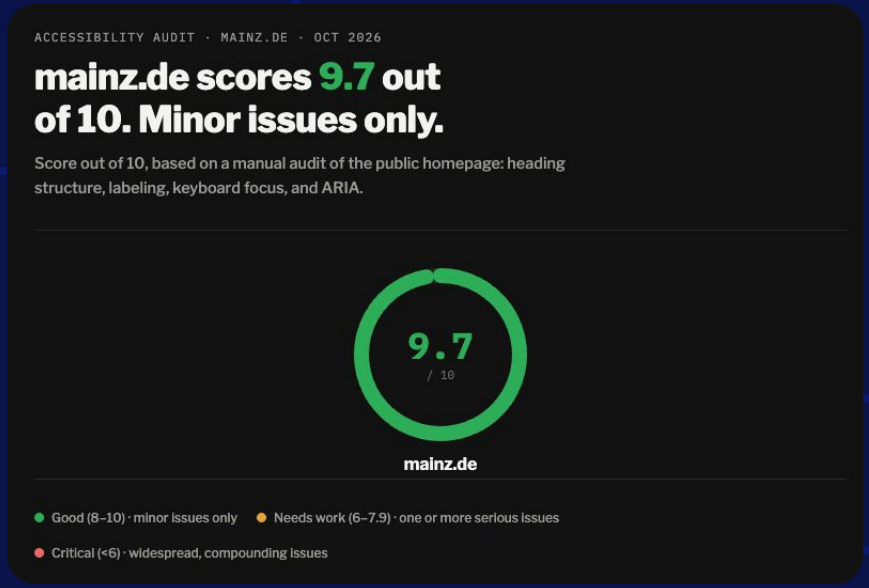
Meet Magento Germany is organized by these three agencies. How accessible are their own sites?

Score out of 10, based on a manual audit of each agency's public homepage: heading structure, labeling, keyboard focus, and ARIA.

customgento.com	e3n.de	run-as-root.de
9.7 / 10	8.7 / 10	6.4 / 10



How about the City?



How do we test for A11Y?

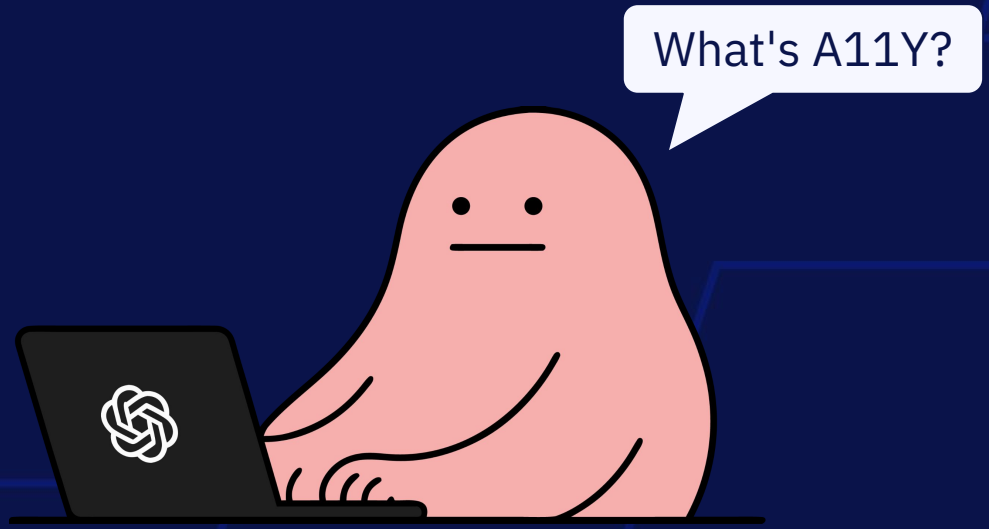
- Manual
- Testing Tools
- AI

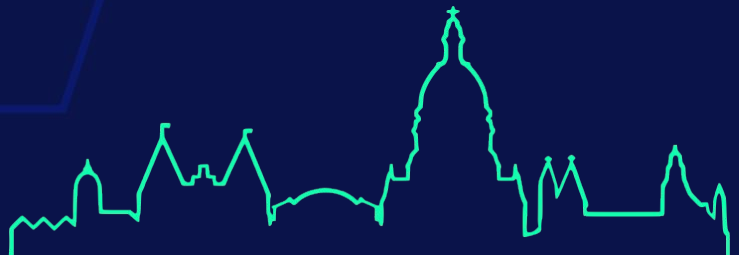


Don’t be a Meat Proxy

Test & Review →

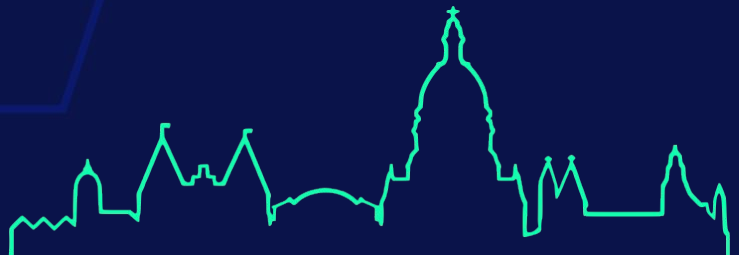
You own the code and the responsibility!





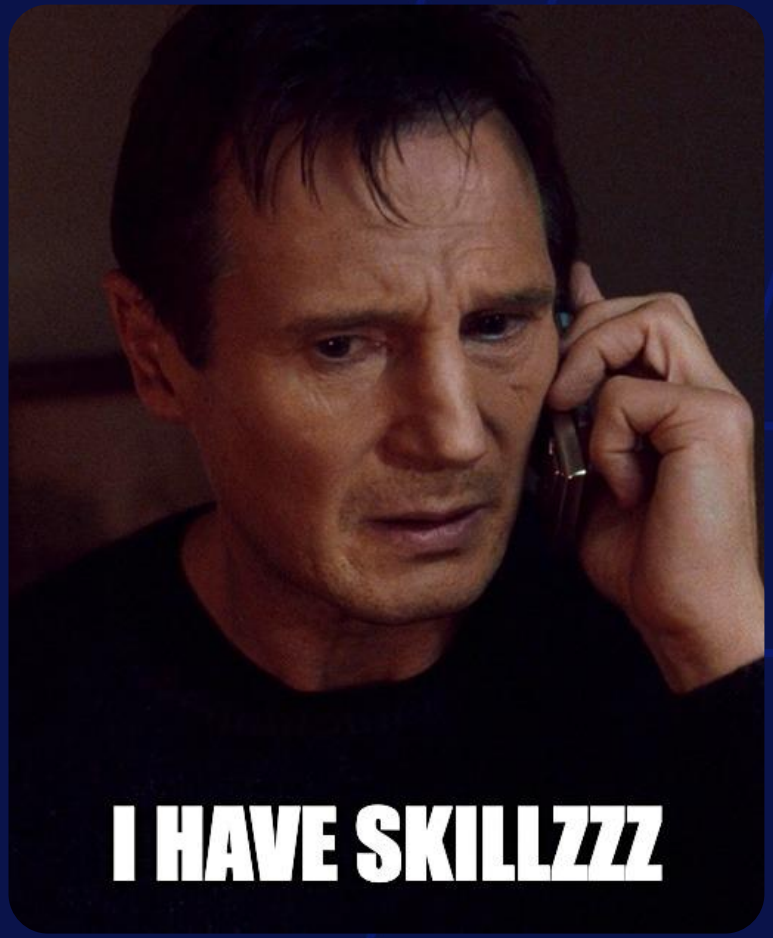
Let's make AI better at A11Y

This is why you're here 👀



SKILLZZZ

- Make a Audit Skill
- Learn your AI, best practises
- Have a UI Library or use one



Making a A11Y Audit skill

STEP 01

Use a MCP that can access your browser

STEP 02

Let it use your browser's devtool

STEP 03

Write a few scripts to do visual testing

```
---
name: a11y-audit
description: "Run a WCAG accessibility audit on a web page. .."
---

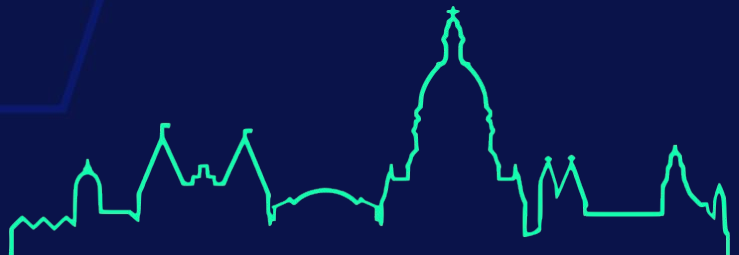
Audit the page with the Chrome DevTools MCP. Work through the steps in order.

If the user did not name a target, assume **WCAG 2.2 Level AA** and say so in the report.

If the Chrome DevTools MCP tools are not available,
stop and tell the user to install it rather than substituting a scripted browser.
Hand-rolled probes are where false positives come from.

The server is [chrome-devtools-mcp](https://github.com/ChromeDevTools/chrome-devtools-mcp),
launched with `npx -y chrome-devtools-mcp@latest`. Registration differs per agent,
so point the user at their own MCP configuration rather than naming a command.

...
```



Learn Your AI: Best Practices

PRACTICE 01

Define Context & Constraints

Provide explicit domain context, WCAG standards, and clear boundaries before generating code to minimize hallucinations.

PRACTICE 02

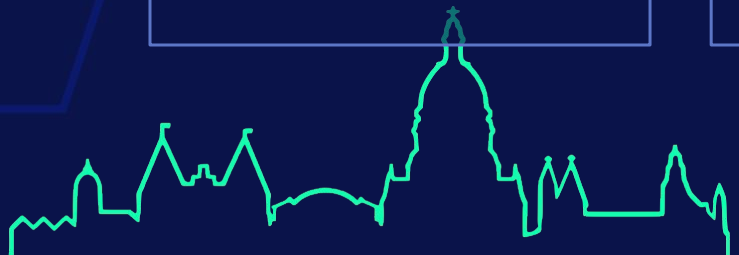
Iterative Refinement

Treat AI output as a draft. Review, test, and feed errors back into the prompt to guide the model toward accessibility compliance.

PRACTICE 03

Human Oversight

You own the code and responsibility. Always perform manual accessibility audits alongside AI generation.



How UI Components Boost Accessibility

BENEFIT 01

Built-in Semantics

Standardized UI libraries ship with native ARIA attributes, semantic HTML elements, and robust keyboard focus management out of the box.

BENEFIT 02

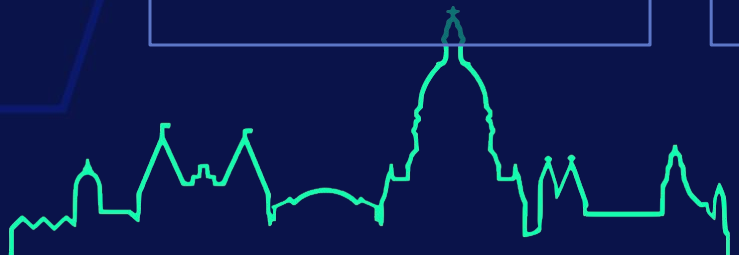
Consistent Patterns

Reusable accessible components ensure predictable keyboard navigation and screen reader behaviors across all application workflows.

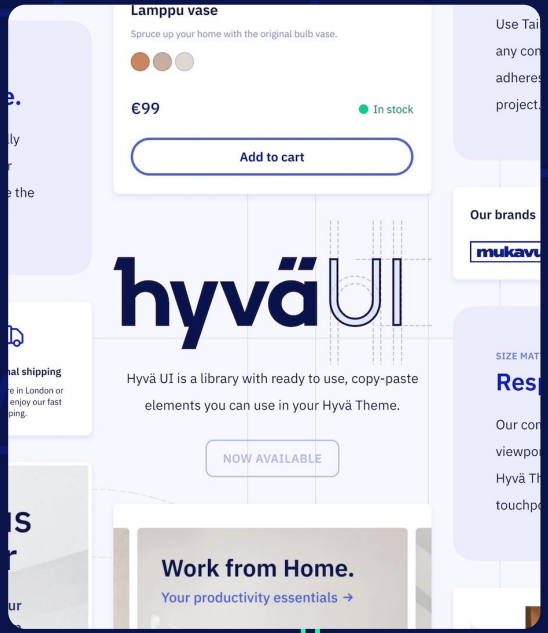
BENEFIT 03

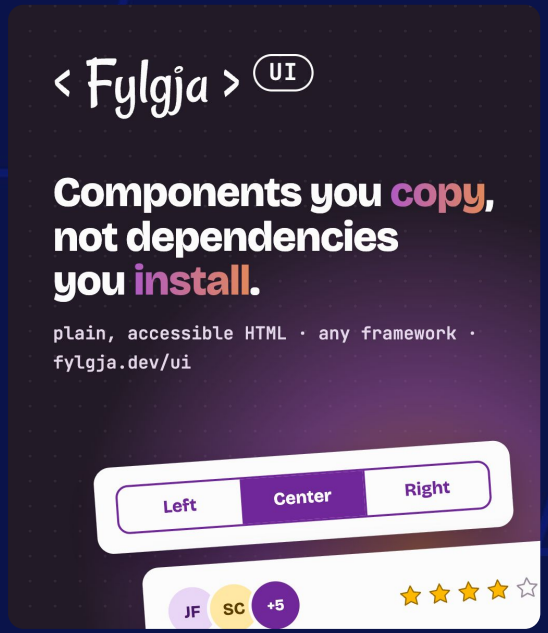
Centralized Auditing

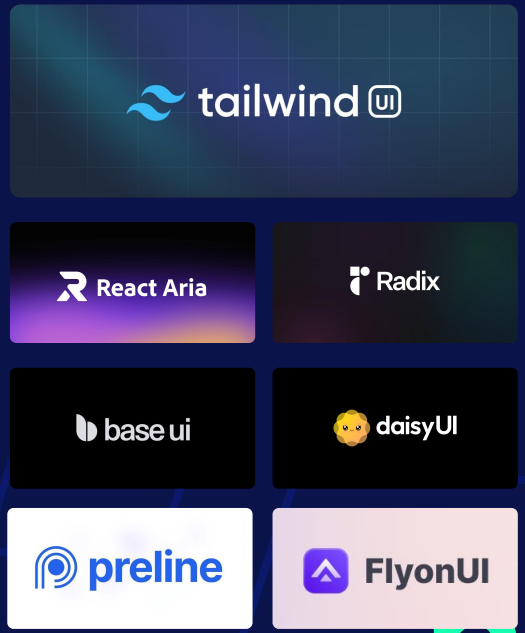
Fixing a single accessibility bug within the component system immediately resolves compliance issues across your entire product ecosystem.



How UI Components Boost Accessibility







< thank you! />

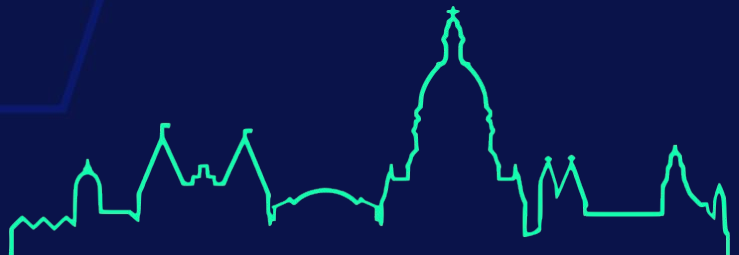
hyva.io

@hyva

@hyva_io



grimlink.com



hyvä