07.

Application accessibility

Making the interface friendly
for every user

Rocky Neurock
Engineering Team Lead at Honeypot.io

> **I do think the key to increased accessibility, and better experiences overall, is for those of us "in the know" to really teach our counterparts about the benefits of accessibility.**

"Don't break the web", my friend Melanie Sumner often admonishes. As developers, we're often last in line to promote accessibility in our work. To escape this pattern, we need a shift in thinking. Accessibility won't come to us – we must become great teachers to our peers so the web can work for everyone.

This topic is near and dear to my heart. Not only because I really care about user experience but also because I suffer from impaired vision. I can say with surety that the web doesn't work for me. Small text and low contrast ratios affect me the most but I routinely encounter other frustrations.

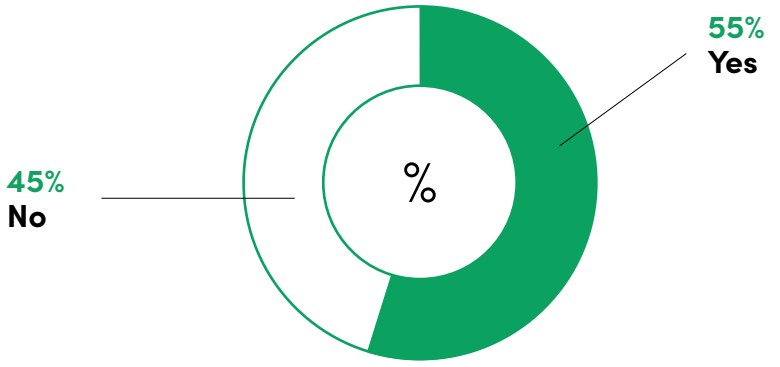
For example, non-native components that reimagine select elements or checkboxes – they can frustrate any user if not done extremely well.

Firstly, consider how nerve-racking these experiences can be for users that prefer getting around the web with their keyboards or for users on mobile devices. Then, think about users with assistive devices. Yes, their experience is even worse.

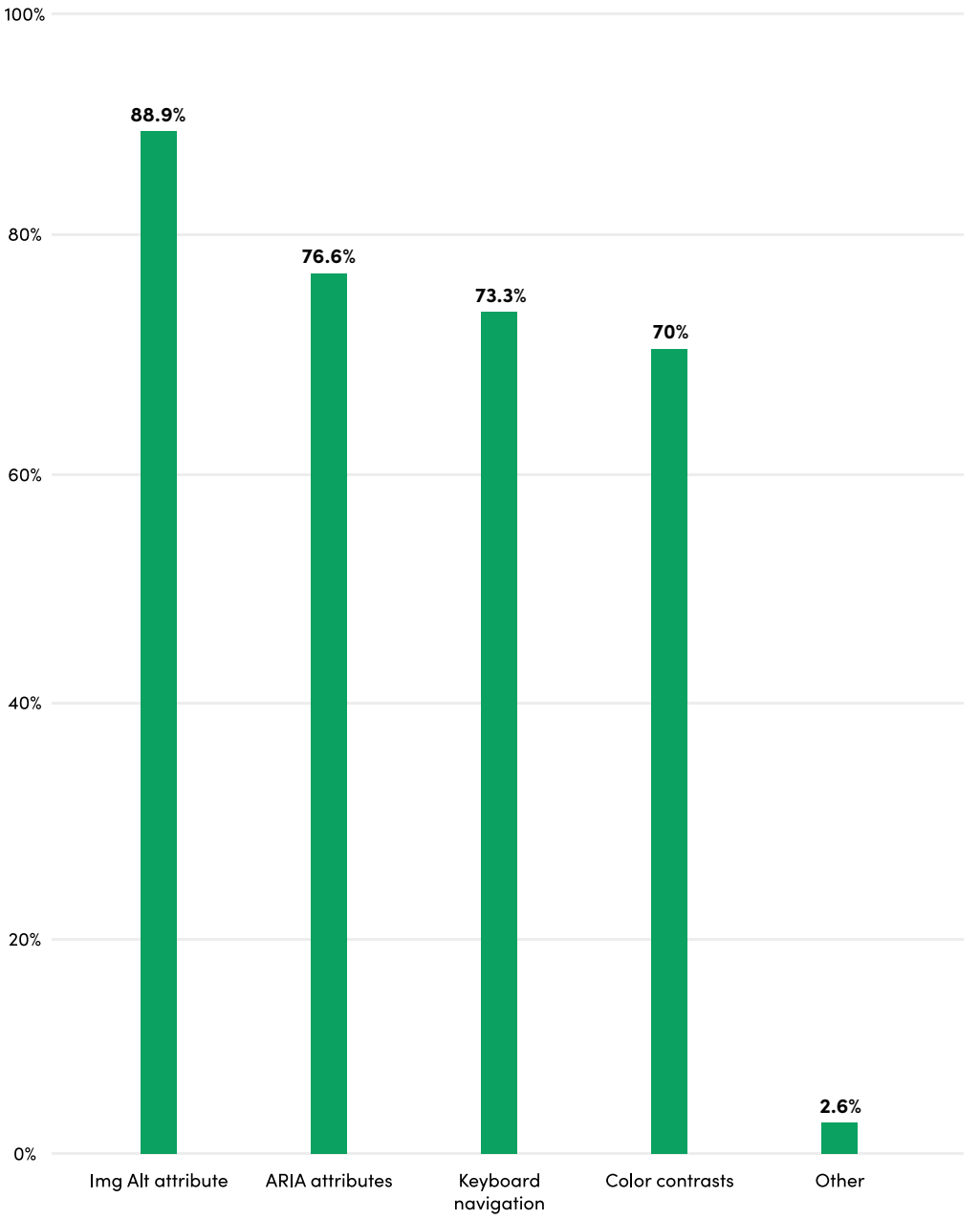
It's good to see that developers who take care of accessibility seem familiar with most of the basic Web Content Accessibility Guidelines (WCAG). In the future, we should also try to find out how many people test for accessibility. There is an ever-increasing number of tools to automatically test accessibility and I wonder if an increase in adoption would correlate to an increase in the percentage of developers who feel responsible for accessibility.

I do think the key to increased accessibility, and better experiences overall, is for those of us "in the know" to really teach our counterparts about the benefits of accessibility. If we can free up some of our own time with automated tests, even better.

Do you take care of application accessibility?



How do you take care of application accessibility?



08.

Development teams

Frontend development?
It’s a team sport

Guillermo Rauch
CEO of Vercel

Recent trends in frontend architecture and deployment infrastructure have influenced how frontend developers collaborate with their team members.

Frontend development is a team sport – shown clearly with 92% of the respondents stating they've worked as part of a development team during the last year. However, recent trends in frontend architecture and deployment infrastructure have influenced how frontend developers collaborate with their team members. At Vercel, we've seen that firsthand.

With the rise of new frontend architectures like Jamstack, frontend developers can deploy the frontend independent of the backend. They no longer have to wait for the full backend test suite to run, resulting in faster iterations. Furthermore, there's a rise in off-the-shelf backend APIs (e.g., headless CMS, identity providers, etc.) that can easily be plugged into your frontend. This enables backend developers to focus more on developing APIs that are unique to the business.

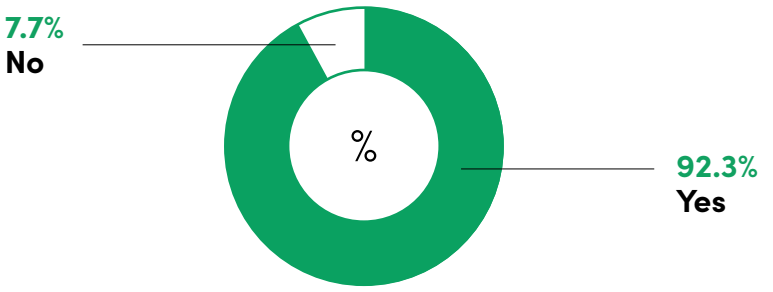
The next change is connected to working with designers and product owners. Because Jamstack apps can be deployed quickly and cheaply

to the CDN edge, it's possible to assign a unique "preview" URL to every branch and every commit. We've done that at Vercel – now, designers and product owners can simply click on the preview URL and instantly see if the changes made by the frontend developer look and work as intended. Much more effective than sharing screenshots and GIFs.

And, finally, software testing. With the introduction of puppeteer, Chrome's headless web browser, combined with serverless compute, end-to-end testing is now fast and cheap. For example, you can have services like Checkly run puppeteer tests — written by QA specialists — against the preview URL. Also, with the rise of Vercel and other frontend deployment platforms which do all the heavy lifting, DevOps engineers can spend less time supporting frontend developers.

Overall, we're very excited about how improvements in frontend architecture and deployment infrastructure are driving changes in developer collaboration. We're looking forward to see more innovations in this space.

Have you worked as part of a development team during the last year?



Which of these people were part of your project development team(s)?