Navigating the Tech World with a Different Lens

Experience of a Visually Challenged Programmer

Arun Kumar
March 2024

I'm Arun, a programmer at Exafluence who happens to be visually challenged. In this rapidly evolving world of technology, where visual elements often dominate, my journey has been filled with unique experiences and insights. I want to share these with you, not just to shed light on the challenges but also to explore how we can collectively work towards a more accessible and inclusive tech environment.

The Day in the life of a Programmer

Imagine a day in the life of a programmer. They start with reviewing emails and project updates, setting their priorities, and planning their workflow. Their day is heavily focused on writing and debugging code, utilizing an Integrated Development Environment (IDE) to craft and test software functionalities. Collaboration is key, involving team meetings, code reviews, and discussions to solve problems and brainstorm. They often consult documentation and online resources like Stack Overflow for technical challenges and learning new methods. A significant part of their day also involves version control and thorough documentation for clarity and future reference.

How does a visually challenged programmer handle these responsibilities

My responsibilities echo those of a sighted programmer, yet the methods and tools I use are markedly different.

Reviewing Emails

I depend on screen readers like JAWS (Job Access with Speech) and NVDA (Non-Visual Desktop Access) for email management, transforming text into speech for effective communication. While composing emails, I use a standard desktop desktop keyboard, as many laptop keyboards lack features like the numpad, which I find essential. Being well-versed in the keyboard layout, I efficiently navigate using tab and other shortcuts. In the Teams mobile app, voice input aids my typing.

Meetings

Participating in verbal discussions is quite natural and easier for me. However, when presentations or code displays are involved, I rely on advance material from my team. Reviewing this material before meetings helps avoid confusion between the screen reader and the live presentation. During my presentations, extra navigation aids are necessary due to the occasional misalignment of screen readers with the content. When fielding questions, I need a moment to align the screen reader for an accurate response. While standard Microsoft and Google software are usually accessible, new releases to applications like Teams, require some getting used to.

Coding and Testing

For coding, I utilize common keyboard shortcuts, finding IDEs like Visual Studio very accessible. Eclipse, enhanced with an NVDA add-on, offers extra shortcuts. Jupyter notebooks pose a challenge with conflicting shortcuts, often leading me to draft in Notepad and then transfer my work. In coding, I navigate character by character or line by line using arrow keys and handle web browser edit boxes and headings effectively. Navigating tables is possible, but charts or markdown insertion is challenging. Alt-texts in images, increasingly prevalent, aid in understanding visual content. Certain external tools, like Postman, are inaccessible, so I find accessible alternatives, like Insomnia for API testing. Traditional debug tools aren't my go-to instead, I read logs and navigate through them, using resources like ChatGPT, Stack Overflow, and Google for troubleshooting.

Documentation and Version Control

For documentation, keyboard usage is standard, though formatting poses some challenges. In GitLab tasks, the command prompt (Git CLI) is my preferred method, and this extends to Python, Docker, and MongoShell, where I opt for CLI over less accessible UIs.

Learning and Staying Updated

Keeping abreast of technological advancements means adapting to continuous changes. I rely on videos and websites for new information. For training, slides and transcripts are my mainstays, and YouTube videos prove useful when the content is verbalized and Github repositories are referred to. Project explanations on GitHub help me understand code details with which I start following the video content.

Conclusion

As you can see, despite the challenges, the adaptations and tools I've embraced not only enable me to overcome obstacles but also make me an effective and proficient programmer. Already there are accessible and inclusive practices and tools in the field of technology, though of course there is always room for improvement. To other visually challenged software developers, let this be an encouragement to pursue your dreams and make them a reality.

If you would like to learn more, write to us at marketing@exafluence.com

Subscribe to our YouTube channel for more solution videos-
https://bit.ly/3FNM0DG

YouTube

For regular updates about Exafluence follow us on LinkedIn https://bit.ly/3FKCqlk

LinkedIn