Free and Open Source Software (FOSS) plays an increasingly vital role in shaping the digital infrastructure of both public and private institutions. Its collaborative model has enabled rapid innovation, transparency, and community-driven development. However, while FOSS aims to be open and inclusive, it often reproduces social inequities found in traditional software development environments, particularly those impacting marginalized groups such as people with disabilities and individuals from diverse linguistic, cultural, and socioeconomic backgrounds. For example, although over 75% of companies rely on FOSS in some capacity, the accessibility and inclusiveness of these tools are frequently overlooked, limiting participation from developers and users who require assistive technologies or face language barriers. This reflects a broader tension within FOSS: its ideals of openness are frequently undercut by structural and technical barriers. This thesis portfolio investigates the question: How can Free and Open Source Software evolve to become more accessible and inclusive, especially for marginalized groups? My technical report proposes a design for an accessible, screen reader–friendly web platform aimed at addressing specific gaps in current FOSS tooling. Meanwhile, my STS research explores how structural and cultural factors within FOSS communities affect participation by underrepresented contributors. Although other challenges, such as global labor inequities and corporate influence on FOSS, fall outside the scope of this work, my portfolio highlights the practical and ethical importance of centering inclusion in both software design and the governance of open-source communities.

The technical report proposes the design of a screen reader–accessible web interface intended to improve participation in FOSS for developers who are blind or visually impaired. Many mainstream code hosting platforms, such as GitHub, provide limited accessibility support for users reliant on assistive technologies. This creates an exclusionary environment, particularly

in spaces where visibility and contribution metrics determine social and technical legitimacy. My design proposal addresses this gap by outlining a platform that adheres to Web Content Accessibility Guidelines and incorporates accessibility-first design principles from the outset. The proposed system uses a modular front-end framework and emphasizes TypeScript for type safety alongside accessibility standards such as semantic HTML, WAI-ARIA labels, and robust keyboard navigation. I identified common pain points in existing platforms through accessibility audits using tools like Axe and Lighthouse and designed UI elements to optimize screen reader interaction. While the platform was not implemented, the report includes detailed wireframes, a feature map, and a development roadmap prioritizing modularity and community-driven extensibility. This design demonstrates that accessibility can be proactively integrated into the software development lifecycle rather than retrofitted as an afterthought. By focusing on inclusive tooling from the design stage, the proposal encourages a broader cultural shift in FOSS development: one that embraces accessibility as foundational rather than optional.

The STS research paper explores how the sociotechnical organization of FOSS projects enables and restricts participation by marginalized groups, especially those with disabilities and individuals from culturally diverse or under-resourced backgrounds. The central research question guiding this work was: Why do FOSS communities, despite their open ethos, remain relatively inaccessible to underrepresented contributors? Using actor-network theory and the Garbage Can Model of decision-making, I examined the dynamics of inclusion, governance, and contribution in FOSS. Through case studies such as GNOME's accessibility initiatives and the FLOSSMetrics project, I identified key structural and cultural barriers. These include the dominance of English in communication, implicit meritocratic norms that disadvantage newcomers, and reliance on unpaid labor, which disproportionately affects contributors from

underrepresented groups. My research also highlighted how some projects have attempted to address these gaps through formalized codes of conduct, mentorship pipelines, and translation tools, but these efforts remain uneven and often tokenistic. Ultimately, the inclusive potential of FOSS remains largely unrealized due to a lack of institutional accountability and an overreliance on volunteer-driven reform. For FOSS to fulfill its democratic promise, accessibility and inclusion must be reframed as core values embedded in both code and community governance, not left to the goodwill of individual contributors.

Together, these projects examine both the conceptual and social dimensions of accessibility and inclusion in Free and Open Source Software. The technical report proposes a tangible, scalable solution to the problem of inaccessible developer tools, while the STS research provides a critical analysis of the sociotechnical systems that perpetuate exclusion. Though neither project aimed to fully resolve the issue, both contribute to a growing body of work that calls for more equitable and intentional design in open-source ecosystems. Future researchers could build on this foundation by implementing and testing the proposed design in real-world communities or by conducting interviews with marginalized FOSS contributors to enrich the qualitative understanding of their experiences. At the intersection of software engineering and social analysis, this portfolio underscores a central conclusion: openness in software is not enough, true inclusivity requires proactive, structural change.