

As Content and Layout Co-Evolve: TANGIBLESITE for Scaffolding Blind People’s Webpage Design through Multimodal Interaction

Jiasheng Li*
 jsli@umd.edu
 University of Maryland
 College Park, Maryland, USA

Zining Zhang*
 znzhang@umd.edu
 University of Maryland
 College Park, Maryland, USA

Zeyu Yan
 zeyuy@umd.edu
 University of Maryland
 College Park, Maryland, USA

Matthew Wong
 mwong119@terpmail.umd.edu
 University of Maryland
 College Park, Maryland, USA

Arnav Mittal
 amittal5@terpmail.umd.edu
 University of Maryland
 College Park, Maryland, USA

Ge Gao†
 gegao@umd.edu
 University of Maryland
 College Park, Maryland, USA

Huaishu Peng†
 huaishu@umd.edu
 University of Maryland
 College Park, Maryland, USA

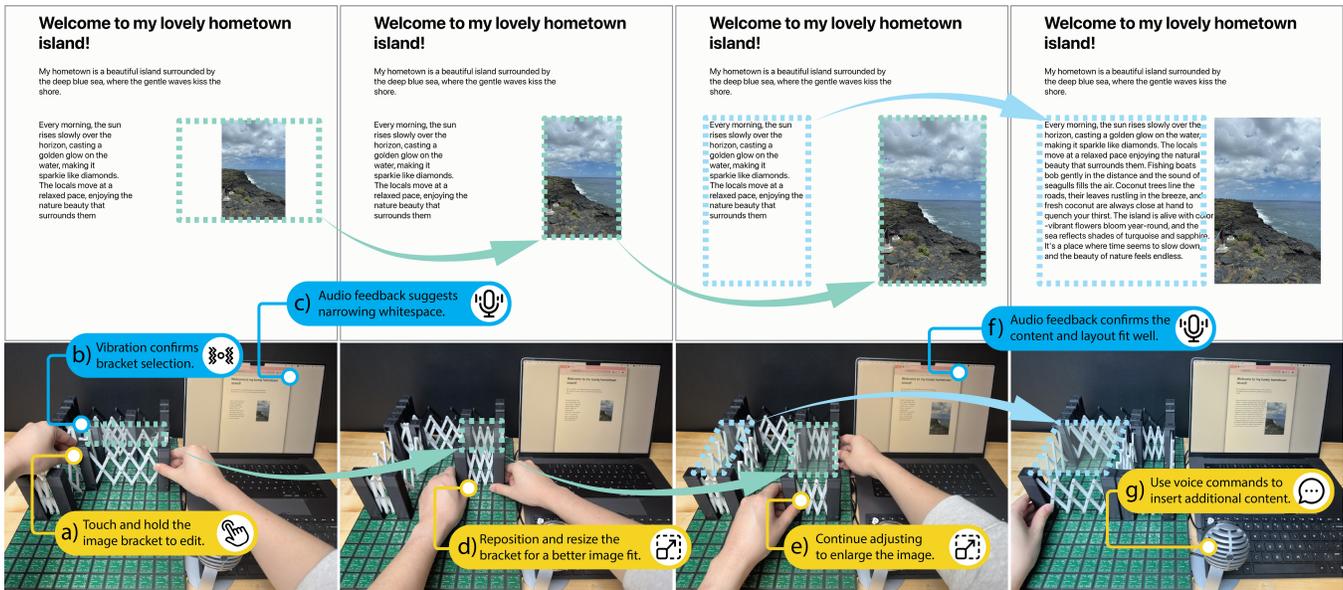


Figure 1: A blind individual iteratively designs a blog webpage using TANGIBLESITE. The top row shows the evolving webpage rendering, while the bottom row shows the corresponding actions performed by the blind designer. The designer a) activates an existing image element (highlighted in mint green) by touching and holding the image bracket, which b) triggers editing mode with vibration feedback. c) Audio feedback detects whitespace on both sides, suggesting the bracket should be narrowed to reduce excess space. The designer d) repositions and resizes the image bracket on the baseboard. The designer e) continues adjusting the image bracket to slightly enlarge the image until f) audio feedback confirms that it fits the layout. The designer then switches to text editing by activating the text bracket (highlighted in light blue) on the left, and g) uses a voice command to insert additional content.

*Co-first authors, equal contribution.

†Co-last authors, equal contribution.



ABSTRACT

Creating webpages requires generating content and arranging layout while iteratively refining both to achieve a coherent design, a process that can be challenging for blind individuals. To understand how blind designers navigate this process, we conducted two rounds of co-design sessions with blind participants, using design probes to elicit their strategies and support needs. Our findings reveal a preference for content and layout to co-evolve, but this process requires external support through cues that situate local elements within the broader page structure as well as multimodal interactions. Building on these insights, we developed TANGIBLESITE, an accessible web design tool that provides real-time multimodal feedback through tangible, auditory, and speech-based interactions. TANGIBLESITE enables blind individuals to create, edit, and reposition webpage elements while integrating content and layout decisions. A formative evaluation with six blind participants demonstrated that TANGIBLESITE enabled independent webpage creation, supported refinement across content and layout, and reduced barriers to achieving visually consistent designs.

CCS CONCEPTS

• **Human-centered computing** → **Accessibility systems and tools.**

KEYWORDS

Accessible web design, multimodal interactions, tangible user interface, visual impairment, accessibility

ACM Reference Format:

Jiasheng Li, Zining Zhang, Zeyu Yan, Matthew Wong, Arnav Mittal, Ge Gao, and Huaishu Peng. 2026. As Content and Layout Co-Evolve: TANGIBLESITE for Scaffolding Blind People’s Webpage Design through Multimodal Interaction. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26)*, April 13–17, 2026, Barcelona, Spain. ACM, New York, NY, USA, 16 pages. <https://doi.org/10.1145/3772318.3791895>

1 INTRODUCTION

Design is, by nature, a process of exploration and refinement. It requires evaluating the current state, identifying areas for improvement, making adjustments, and then reassessing the results. In web design, this often takes the form of continuously reviewing the visual arrangement of elements and considering how their layout and composition shapes the overall experience.

For sighted designers, the iterative cycle is greatly supported by rapid prototyping tools that provide instant visual feedback (e.g., Figma) or coding environments with built-in previews (e.g., Visual Studio Code). For blind individuals, however, perceiving a webpage, identifying differences, and applying changes is undeniably challenging. While screen readers can support experienced and highly-skilled blind designers in website creation [19, 37], these tools convey limited spatial information about web elements and provide limited access to visual formatting [74].

In response, HCI researchers have been developing various assistive technologies to lower barriers to web design for blind individuals [46, 56, 66]. For example, Li et al. introduce a system that uses embossed tactile templates overlaid on a tablet to help blind

individuals explore and design spatial webpage layouts [45]. TangibleGrid [46] provides a tangible interface for real-time webpage layout editing. DesignChecker [25], a browser extension, enables blind web developers to compare their designs with online references and edit HTML or CSS files. While these systems advance support for creating and inspecting graphical web representations, they provide limited support for an end to end web design workflow in which a webpage can be created from scratch and iteratively refined through coherent editing of both content and layout.

In this work, we aim to fill this gap by enabling blind individuals to create and iteratively refine webpages through integrated editing of content and layout. To ground this goal in blind designers’ strategies and support needs, we first conducted two rounds of co-design sessions with three blind participants using a tangible design probe inspired by TangibleGrid [46].

In these sessions, we observed that participants preferred content and layout to co-evolve, refining both in tandem when possible. When designs were simple, they alternated naturally between editing content and rearranging layout elements (Figure 2). As layouts became more complex, however, this flexibility diminished and participants expected more external cues to stay oriented (Figure 4). This suggests a need for systems that preserve structure across edits so blind designers can verify changes without reconstructing the page from memory. We also observed that touch played a central role in inspecting elements, confirming system state, and planning next steps (Figure 6). Because similar touch actions often served different functions, touch interactions should clearly distinguish intents such as inspecting, selecting, editing, and moving through touch coupled feedback. Participants additionally expected explicit, actionable audio feedback on content fit and element relationships within the layout.

Based on these findings, we designed TANGIBLESITE, an accessible web authoring environment that supports webpage creation as content and layout co-evolve (Figure 1). To reduce blind designers’ memory burden, TANGIBLESITE maintains an up-to-date representation of the page’s structure and content. It provides real-time multimodal feedback at both the element and page levels, supporting refinement of individual elements and understanding of the overall relationship between content and layout. To evaluate TANGIBLESITE, we conducted a formative study with six blind participants. All participants created a webpage and iterated on both layout and content. They reported increased control over the design process and relatively low frustration during authoring.

In summary, our paper contributes: (1) an empirical investigation of blind designers’ preferred workflow for integrating content and layout in web design; (2) TANGIBLESITE, an accessible multimodal web authoring prototype that supports real-time feedback and iterative refinement of both; and (3) a formative evaluation with six blind participants demonstrating end-to-end webpage creation and refinement.

2 RELATED WORK

2.1 Nonvisual Access to Graphical Interfaces

A substantial body of work has explored how audio and haptic modalities support blind people in navigating and understanding existing graphical interfaces. Early research, such as Sonic Grid [28],

uses auditory icons and speech feedback to support navigation within digital interfaces. Gesture based overlays that integrate with screen readers such as VoiceOver and JAWS provide additional ways for blind users to access on-screen elements [32, 33].

Although these approaches enable effective access to text, they offer limited support for understanding spatial or graphical structures. To address this gap, researchers have introduced accessibility software [43, 59, 60], sonifications [73], and auditory cueing strategies [8, 9, 14, 39] that describe or encode visual information. ImageExplorer [43] uses a suite of deep learning models to generate multilevel image descriptions, while VoxLens [60] supports querying graphs, tables, and charts. EdgeSonic [73] conveys object boundaries on touchscreens by encoding distance to edges through pitch and timing. Additional work has extended nonvisual access to maps [10, 61, 68]. For example, Jimenez et al. [10] propose an SVG encoding of indoor layouts that enables screen readers to verbalize spatial structure.

In parallel, haptic and tangible interfaces have been explored to complement these auditory techniques. Vibrotactile patterns have been used to encode geometric shapes [3, 53]. Tactile graphics and refreshable tactile displays provide representations of textures, mathematical content and graphs [5, 17, 30, 41, 50, 51, 65, 70, 72]. Other work has introduced 3D printed tactile artifacts for maps [15, 22, 23] and graphs [12]. TouchPlates [34], for example, uses physical overlay on touchscreens to help blind people locate interactive virtual buttons.

Taken together, this body of work provides extensive solutions for interpreting existing visual interfaces. However, the focus has largely remained on access and interpretation rather than on supporting blind individuals as creators of visual or spatial content.

2.2 Visual Content Authoring Tools for the Blind People

Alongside work on nonvisual access, researchers have developed systems that support blind individuals as creators of visual media, as blind individuals have expressed strong interest in producing drawings, photographs, and other visual artifacts [74]. Early tools such as TDraw [40] enable nonvisual drawing on swell paper with audio feedback. More recent drawing systems such as AudioDraw [20] and A11yBoard [77, 79] use gestures, vibrotactile cues and auditory descriptions to help blind individuals construct geometric shapes on digital artboards. Tools such as VizSnap [1] and EasySnap [29] support blind individuals in taking photographs by providing audio guidance for framing.

Recent progress in AI has broadened what blind creators can produce. A11yShape [78] assists blind users in creating 3D models. GenAssist [26] provides guided image generation and refinement. AVScript [27] supports nonvisual editing of video clips. AltCanvas [44] decomposes an image into tiles and uses AI to generate or edit each tile before recombining them into a complete image. Across these systems, language and vision models help blind individuals describe, generate, and edit visual media with control.

Blind creators also face barriers in evaluating visual outcomes or understanding visual concepts, and therefore frequently rely on sighted collaborators to assess their designs [74]. Diffscriber [55] helps blind slides creators understand changes made by sighted

collaborators to both slide content and layout by automatically identifying and describing those edits. To further support blind creators in assessing their own slide designs, Zhang et al. [76] build a large dataset to support automated slide evaluation. DesignChecker [25] analyzes webpages created by blind programmers and provides code-level suggestions to refine layout and content. VizXpress [75] assesses photographic aesthetics and provides editing guidance.

Despite these efforts to support blind people in drawing and visual content editing, most systems operate within a single unit of content (e.g., one image, one photo) rather than addressing multi-element graphical designs. In this paper, we instead focus on entire webpages composed of multiple elements such as text paragraphs, images, and videos, and explicitly investigate how to help blind designers identify and understand content–layout issues. TANGIBLESITE gives blind designers control over the spatial organization of these elements, not just the content itself.

2.3 Accessible Web Creation

Creating webpages traditionally requires knowledge of HTML, CSS, and JavaScript. Considerable effort has thus been devoted to better equipping blind individuals interested in web programming through workshops, bootcamps [35–38], and online courses [11, 54]. Yet, programming tools themselves often remain inaccessible [2, 21, 24, 49]. For example, screen readers do not fully convey code structure, leading to over reliance on memory [48]. To address these issues, researchers have developed various tools to improve code understanding [4, 16, 18], navigating in-line locations [57, 58], and debugging [6, 66, 67]. While effective for developers who are comfortable with code, these approaches remain challenging for many novices or blind individuals who prefer non code based design workflows.

The primary approach to codeless web design is through graphical web authoring tools such as Figma, WordPress, and Google Sites. More recently, advances in generative AI have enabled intent-driven and example-driven authoring workflows [13, 42, 63, 64, 69, 71]. One such example is MISTY, a workflow that leverages LLMs to help designers rapidly retrieve online examples within ongoing designs, accelerating exploration and iteration [47].

Despite this progress, most graphical web design tools, whether traditional or more recent, are not designed with blind individuals in mind, and few offer nonvisual feedback. Complex layout information combined with page content is difficult to convey through standard screen readers.

To lower barriers to web design for blind individuals, recent HCI research has explored nonvisual design approaches such as tangible representations, tactile printouts, and touchscreen-based design systems. Li et al. [45] introduce a system that prints tactile webpage templates to overlay on a touchscreen, enabling blind individuals to explore page structure and perform layout edits through touch and audio. While effective for conveying spatial relationships, the need to regenerate and reprint tactile sheets after each edit slows the feedback cycle and can make refinement challenging for complex or frequently changing designs. TangibleGrid [46, 62] introduces a physical baseboard with shape-adjustable brackets to represent layouts, enabling direct creation, adjustment, and deletion of elements with real time feedback. Although intuitive and spatially

Table 1: Comparison of systems that support accessible webpage design.

System	Tangible Layout Editing	Content Creation and Editing	Real Time Feedback for Iteration	Content–Layout Fit Guidance
Potluri et al. [56]	×	△	×	×
Li et al. [45]	✓	×	×	×
TangibleGrid [46]	✓	×	△	×
TANGIBLESITE (ours)	✓	✓	✓	✓

grounded, it focuses solely on layout structure and does not support content editing, which can make full webpage authoring less seamless. Other systems, such as the one introduced by Potluri et al. [56], use touchscreens to edit element attributes (e.g., padding, font size) and automatically update CSS source files. This approach provides control over styling, but its emphasis on attribute level adjustments offers less support for understanding or manipulating the larger layout structure of a webpage. Across these efforts, the focus has largely remained on layout representation and manipulation, with few tools supporting content entry alongside layout design or providing real time feedback for both during design.

Our TANGIBLESITE builds on this prior work by enabling blind individuals to create webpages through multimodal interaction. Unlike previous systems, TANGIBLESITE supports content and layout design as they co-evolve by coupling tangible layout manipulation with audio feedback. Tangible interaction provides a spatial scaffold for externalizing and revisiting layout decisions, while audio feedback evaluates content fit, signals layout conflicts, and suggests concrete adjustments. Table 1 summarizes the key differences between TANGIBLESITE and previous systems.

3 UNDERSTANDING BLIND INDIVIDUALS' INTERACTIVE WORKFLOWS

Designing a webpage involves coordinating content and layout, often in ways that depend on nuanced visual cues such as proportion and spatial alignment. For blind individuals, the absence of these visual affordances makes the process significantly more challenging, especially when it involves multiple elements and when iterating between local and global decisions. To investigate how blind individuals conceptualize and navigate content–layout relationships, we conducted two rounds of co-design sessions using a tangible design probe modeled after TangibleGrid. Our aim was to surface both the challenges blind individuals face and the interaction mechanisms needed to support them. Our study was guided by the following research questions:

RQ1. How do blind individuals conceptualize and organize the workflow of content and layout during webpage creation?

RQ2. What types of information help blind individuals detect conflicts between local content and layout, and maintain page-level consistency during iteration?

RQ3. What features and mechanisms should authoring tools provide to help blind individuals resolve global layout consequences of local edits?

These questions shaped the design of our co-design sessions and served as the analytic structure for the findings below.

3.1 Apparatus

We used a simplified version of TangibleGrid as our primary design probe. The probe consisted of a 3D-printed tactile baseboard whose groove structure represented the underlying grid of a webpage, paired with a set of resizable brackets representing web elements such as text blocks, images and videos. Brackets included tactile patterns indicating element types and could be resized by repositioning their corner pillars. This allowed participants to physically inspect size, location, and spatial relationships.

Unlike the full TangibleGrid system, our probe did not include audio or computation. Instead, one researcher acted as the system, interpreting participants' spoken commands, simulating audio feedback, and narrating digital changes. This low-fidelity approach ensured that participants' strategies emerged naturally without being shaped prematurely by system assumptions. A keyboard was provided for participants who preferred typing over voice input.

3.2 Participants

We recruited three participants through the local National Federation of the Blind (NFB) email list [52], and all completed both co-design sessions. Participants self-reported being blind and self-identified as women (ages 19–69, $M = 43$, $SD = 20.4$).

The purpose of the co-design sessions was to surface design practices that benefit from deep, contextual observation. We therefore worked closely with a small number of participants in extended sessions that involved layout construction, content entry, and iterative refinement with the tangible probe. This format supported close observation of participants' choices and strategies and produced detailed behavioral and decision-making data. We summarize participants' prior design experience below.

P1 had no prior experience with webpage design but had created 2D graphical content such as PowerPoint presentations. P1 typically used predefined templates for text and layout, rarely added images or modified visual appearance, and consistently relied on assistance from a sighted collaborator to verify the design.

P2 was actively involved in a project to develop a website focused on aging and disability consulting. P2 was responsible for gathering the necessary content for the site, but did not directly implement the webpages herself. Instead, P2 collaborated with a sighted partner who handled the web design tasks, including layout design and translating P2's ideas into the final implementation.

P3 had experience developing websites in a Linux environment. P3 noted that the process required extensive memorization, particularly of coding syntax. Despite the availability of accessible tools for blind developers, P3 still faced considerable challenges, such

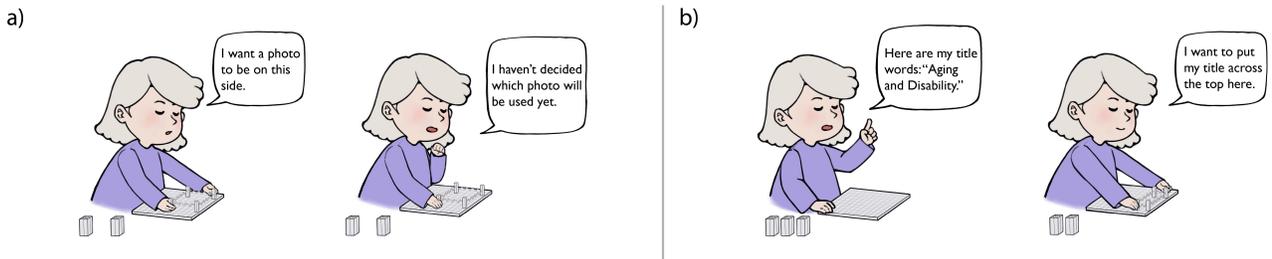


Figure 2: P2 demonstrated two different workflows. a) Placed the bracket first to establish the layout, then inserted the content. b) Spoke the content first, then placed the bracket to determine its placement.

as limitations in the comprehension of webpage layouts and the adjustment of web elements' dimensions.

This study received approval from the university's Institutional Review Board (IRB), and each participant was compensated at a rate of \$35 per hour.

3.3 Data Analysis

All sessions were video and audio recorded with consent. We analyzed the data using a qualitative approach that combined video interaction analysis [31] and thematic analysis [7]. We first transcribed all spoken dialogue, including participants' questions, reasoning, and commands, as well as the simulated system responses. In parallel, we reviewed the video recordings to document nonverbal behaviors such as hand movements, interaction sequences with the tangible probe, and strategies not evident from speech alone. We then conducted thematic analysis, iteratively coding transcripts and video notes to identify patterns in how participants conceptualized content and layout, detected conflicts, and responded to feedback. Codes were refined through repeated comparisons across participants and between co-design rounds, resulting in a set of consolidated themes that informed our design implications.

3.4 Co-design Session 1

3.4.1 Procedure. After a brief interview about participants' past experiences, we introduced our design probe and provided two design scenarios. Scenario 1 focused on a single-element webpage design. Participants were asked to create one webpage element of their choosing using the probe. This scenario isolated basic operations and allowed us to observe how blind participants approached layout and content when no constraints were imposed. Scenario 2 involved designing within a more complex, multi-element layout. Participants were asked to add new content or modify existing elements within a crowded spatial arrangement involving multiple brackets. This scenario reflected more realistic design challenges in which multiple components interact and spatial constraints matter.

At the end of the session, we conducted a semi-structured interview with open-ended questions focusing on participants' design rationales, strategies for navigating and interpreting multiple elements, and challenges encountered during content creation and layout editing.

3.4.2 Findings. We report three findings (F1–F3) across two co-design sessions. This section presents only F1 from Co-design Session 1; Co-design Session 2 reports F2 and F3.

F1. Participants used flexible workflows in simple designs, but this flexibility diminished as complexity increased. Co-design Session 1 shows that blind participants value flexible workflows during webpage creation, readily interweaving content creation and layout composition when the design is simple. However, once multiple elements are involved, this flexibility becomes much harder to sustain, and participants shift toward a layout-first strategy to manage spatial complexity.

In the single-element scenario, participants demonstrated considerable freedom in how they initiated design. Some began by placing a bracket first to anchor a spatial position, then described the content they wanted to associate with it (Figure 2a). At other times, the same participant first thought through and verbalized the content, then placed the matching bracket afterward (Figure 2b). In this single-element context, participants were able to iteratively refine their content. For example, during one design iteration, P1 expanded a short text description by enlarging the corresponding bracket to create space for additional content. P1 highlighted the flexibility to adjust both structure and content:

"Once you broaden the bracket, you can go back to adding content. That's a good thing, because sometimes you get an idea, and then you can change the bracket and go back. I feel like that's what I can do with the baseboard and brackets."—P1

This flexibility, however, diminished as the webpage grew more complex. When multiple elements were involved, as in the scenario 2, participants no longer began with content. Instead, they prioritized establishing and verifying the layout. They spent time touching existing brackets to understand the current structure, then placed new brackets to secure the position and size before entering content. For example, in one session, P2 laid out four brackets in advance to establish locations for the elements they envisioned (a title, two images and a paragraph) and then returned to fill in the content for each (Figure 3). Similarly, P3 placed brackets for three elements in sequence, explored the board again to confirm spatial relationships, and only then added content (Figure 4a-d). P3 later added the final bracket in the same manner, establishing the layout before inserting the media content (Figure 4e-f).

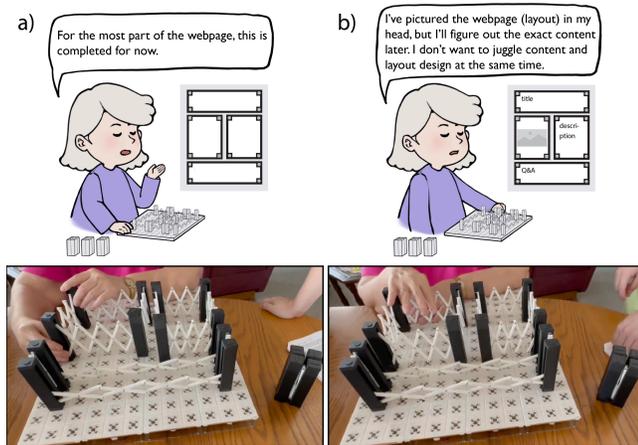


Figure 3: a) P2 used multiple brackets as placeholders to sketch the overall webpage structure. b) P2 then discussed the intended content for each bracket.

We also observed fewer edits in complex scenarios. Participants continued adding new elements but rarely revisited or modified earlier ones. As the number of elements increased, changing either content placement or layout relationships demanded re-establishing the mental model of the entire page, which became cognitively taxing. This applied to both spatial adjustments and decisions about how newly added content should relate to existing elements.

Overall, these findings suggest that nonvisual webpage design imposes growing cognitive demands as complexity increases. Therefore, rather than supporting only layout or only content, systems should help blind designers maintain a clear understanding of both the evolving content and the page's structure. Tools should help offload memory, provide summaries of what is present, highlight relationships among elements, and offer timely feedback that reduces the cost of reorientation. This raises the question of which

types of feedback best support blind individuals in managing both content and layout as designs grow in complexity, which in turn motivated the focus of our Co-design Session 2.

3.5 Co-design Session 2

Building on insights from Co-design Session 1, we investigated how targeted system feedback could help blind individuals detect and negotiate content–layout conflicts. We presented participants with four common types of misalignment that occurred across text, image, and multi-element configurations (Figure 5) and examined (1) what feedback was most helpful for participants in identifying these issues, (2) when such feedback should be delivered, and (3) how participants preferred to respond once a conflict was identified. Our broader aim was to understand how multimodal feedback can support blind designers in perceiving the current design state and maintaining an accurate mental model of the webpage.

3.5.1 Design scenarios. We introduced four scenarios that exemplified typical challenges blind individuals face when composing webpages using bounded containers.

The first two scenarios focused on text editing. Because webpage text must fit within bounded regions, participants encountered two opposite challenges: text overflow, where content extended beyond the container's bounds (Figure 5a), and text underflow, where the container was much larger than the content, leaving substantial whitespace (Figure 5b). These issues are easy for sighted designers to notice at a glance but are much harder to perceive nonvisually.

The third scenario focused on image–container mismatch (Figure 5c). Even when inserting an image was straightforward, blind individuals could struggle to assess whether the container's shape aligned with the image's aspect ratio and whether the result involved unintended cropping or excess unused whitespace.

The fourth scenario addressed multi-element adjustment (Figure 5d). When attempting to add text to a filled container, blind individuals may find that neighboring elements constrained further expansion. This introduced a need to reposition or resize multiple

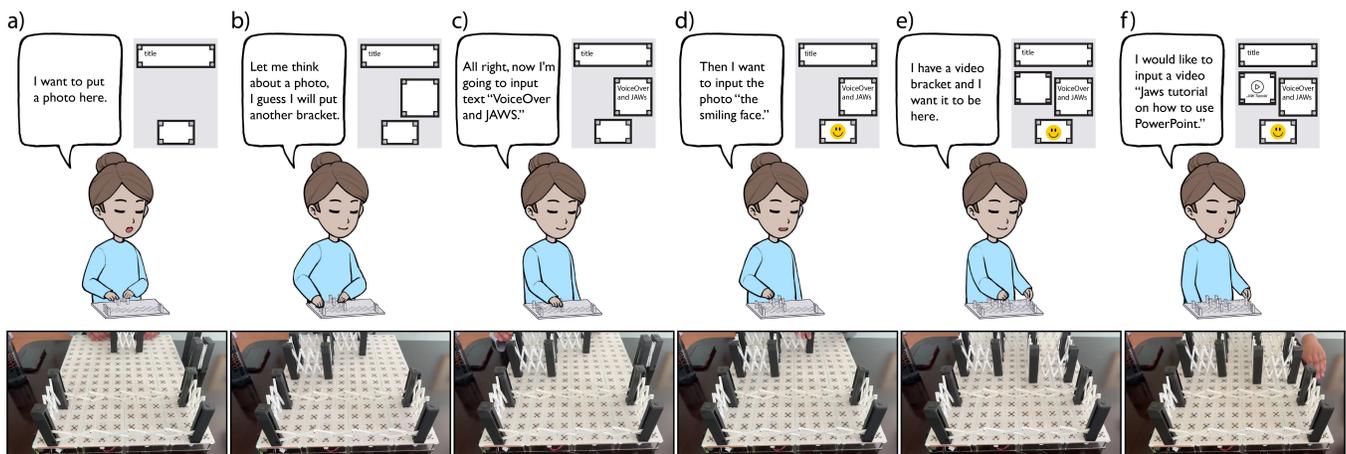


Figure 4: Workflow of P3 designing webpages using multiple web elements. a-c) P3 placed three web elements in sequence. d) P3 explored the board to verify the alignment and then inserted content into each bracket. e-f) P3 added an additional bracket on the baseboard and input its content.

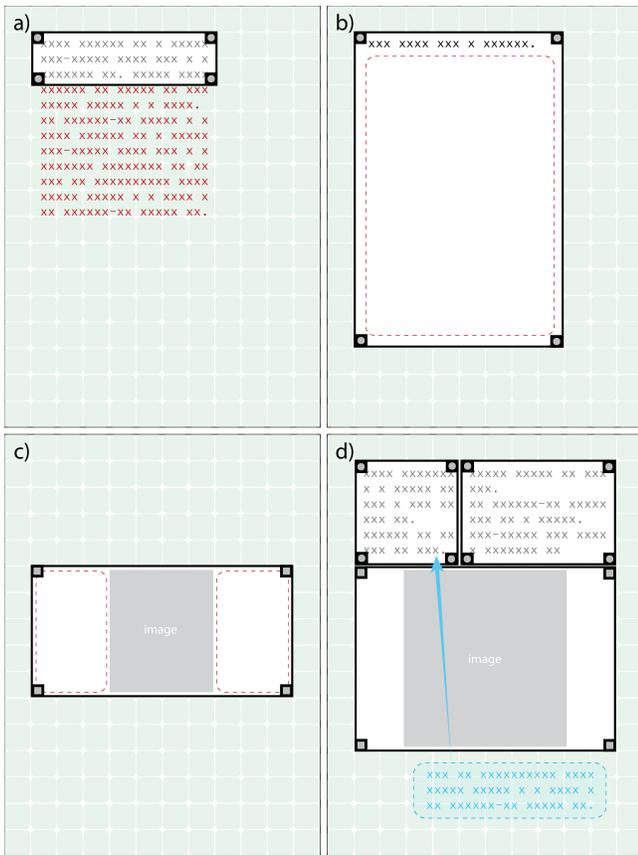


Figure 5: Four scenarios examined in Co-design Session 2: a) **Text overflow:** the text container is too small, causing content to extend beyond its boundary. b) **Text underflow:** the container is larger than the text, resulting in excessive unused space; the dashed inset marks the usable text area. c) **Image-container mismatch:** a landscape-oriented container holds a portrait image, leaving whitespace on the left and right. d) **Multi-element reorganization:** when an existing text container is full, adding new content requires resizing or repositioning neighboring elements to create space.

elements simultaneously, which proved challenging without clear awareness of the underlying spatial relationships.

Together, these four scenarios represent common design states where nonvisual webpage authors may lose awareness of how content and layout interact.

3.5.2 Procedure. Co-design Session 2 was conducted on a different date from Co-design Session 1, with all three blind participants from the first session participating in this follow-up study. Each session began with an explanation of the four scenarios and the study goals. Participants worked through each scenario while offering feedback on how the system should help them detect content–layout conflicts and decide how to resolve them. One researcher role-played the system, providing audio feedback and clarifications, while another assisted with manipulating the tangible brackets when needed.

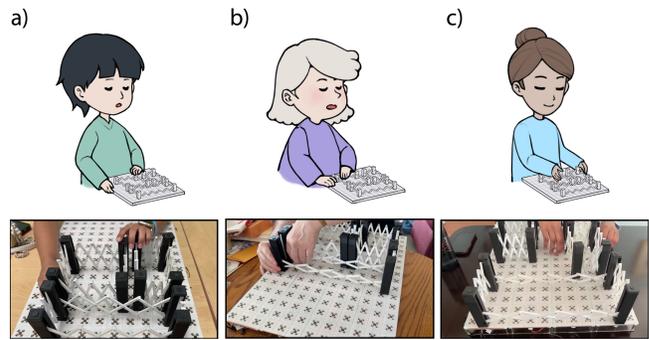


Figure 6: Examples of three touching behaviors. a) P1 used touch to identify the category of a bracket by feeling the tactile patterns on its top. b) P2 used their hands to touch brackets, the baseboard, and tactile patterns to check alignment and gauge size. c) P3 swept the board and used their hands to touch multiple brackets to recall layout decisions.

We observed participants’ strategies, probed their reasoning with follow-up questions after each scenario, and concluded with an open discussion of preferred feedback types and timing. Each 90-minute session was video- and audio-recorded, transcribed, and anonymized for analysis.

3.5.3 Findings. We present findings from Co-design Session 2. We continue the numbering from Session 1.

F2. Hands as the primary channel for monitoring, confirming, and planning. Across all scenarios, participants relied on their hands as the primary channel for understanding and reviewing their webpage designs (Figure 6). For example, when participants lost track of the category of a bracket, they used touch to perceive the tactile patterns on the top of the bracket to remind themselves of its category. When they needed to check alignment, gauge the size of a bracket, or simply recall their layout decisions, they naturally used their hands to touch the bracket or trace the grooves on the baseboard. Participants also swept their hands across the design canvas to refresh their mental model of the layout and keep track of their overall design status. The grooves and tactile features embedded in the tangible interface helped them stay aware of the layout and prompted them to think about the corresponding content.

Hands were equally important for planning future actions. Participants frequently kept a hand on a bracket while deciding on their next steps. When referring to a specific web element, they always placed a hand on that bracket or pointed to it to ensure they were talking about the correct element on the design canvas. After placing a bracket, they kept a hand on it while confirming whether it was operating in the content input mode and able to accept the intended content.

This reliance on hands for both monitoring and action underscores the need for systems that support hand-based interaction as a primary channel. At the same time, systems should clearly signal the current interaction mode and help disambiguate user intent so that touch-based exploration does not lead to confusion or unintended actions.

F3. Preference for actionable audio feedback on content–layout fit. When hands were not sufficient to identify or resolve design challenge such as content–layout misfit, participants turned to audio support. However, most blind participants did not simply ask for “more description.” Instead, they articulated specific ways audio feedback should reveal the content–layout relationships and help them evaluate whether their design choices achieved the intended effect.

When entering text, participants may struggled to judge whether the content would fit within the bracket’s available space. They asked questions such as “How many words did I insert?” and “How many words can this bracket hold?” In one text–entry instance, P2 paused mid–sentence to count words and reconsider their phrasing, unsure whether the text would fit within the bracket. She explained, “I’m quiet because I’m just counting the words (in the text content, and thinking about) how I want to express this.” P1 similarly asked whether touching a bracket could report the number of words it contained. These behaviors suggest that understanding content capacity is cognitively demanding without explicit feedback. Participants wanted the system to provide bracket capacity and remaining space so they could decide whether to edit the text or adjust the layout rather than rely on their own memory or guesswork.

Participants described similar challenges with media elements. Choosing an image and inserting it was accessible with the keyboard, but understanding how the image was displaced within the bracket was not. They requested information about orientation, aspect ratio, and the amount of whitespace surrounding the image. P1 explained, “It does matter if the picture is vertical or landscape. I don’t know how the picture views in the bracket unless I ‘see’ it.” Participants also wanted the system to suggest concrete adjustment options in terms of rows and columns they could physically trace with their hands. As P3 noted, “I want it (the system) to tell me and to recommend what I should do, if I should extend it by width or height, because I really don’t want whitespace in my image bracket.”

Participants also asked for feedback that summarized the design at a broader level. While they could touch individual brackets to recall type and content, they still struggled to keep track of empty regions or gaps on the canvas. P2 explained that they wanted feedback on where unused space remained so they could avoid leaving “too much whitespace,” which would reduce visual appeal.

Taken together, these observations indicate that audio feedback should provide actionable information that goes beyond simple descriptions. Systems need to report how content fits within available space, highlight spatial relationships, and offer concrete options for resolving mismatches so that blind designers can make informed decisions about their layouts.

3.6 Design Opportunities

Drawing on findings F1–F3 from the two rounds of co–design, we identified three opportunities that directly address how blind individuals coordinate content and layout, use touch as a multifaceted interaction channel, and rely on actionable guidance to manage growing design complexity. These opportunities guided the development of the TANGIBLESITE system.

D1. Maintain a persistent, up–to–date representation of the evolving page. Participants frequently shifted between content–first and layout–first strategies, but this flexibility became difficult to sustain as more elements were added. To support content and layout co–evolving during design, tools should maintain an updated model of all content and layout information and make it accessible on demand.

D2. Provide immediate, touch–aligned feedback that clarifies system state and user intent. Touch served many roles at once: inspecting past decisions, confirming current actions, and planning future edits. Systems should respond to touch with localized haptic cues that confirm selection, mode, and the status of the element being handled. Feedback needs to be sensitive to the context of touch so that review, editing, and exploration can stay risk–free and predictable.

D3. Offer actionable audio feedback about content–layout relationships. Participants did not simply want a description. They wanted feedback that explained how content fit inside containers, revealed mismatches in image orientation or text capacity, and suggested concrete adjustments they could choose from. Tools should provide concise structural audio guidance, including state summaries and recommendations for resolving conflicts across elements. This support scales from single–element edits to multi–element layouts, where participants struggled most to maintain a coherent mental model.

4 TANGIBLESITE

TANGIBLESITE is an accessible web authoring tool for blind designers the joint development of content and layout. It maintains an up–to–date representation of the page’s structure and content, and provides real–time multimodal feedback via tangible, auditory, and speech–based interactions. Figure 7 shows the overall system setup.

4.1 System Walkthrough

To illustrate how TANGIBLESITE supports nonvisual webpage design, we follow Zoe, a blind college student interested in web development with no prior coding experience. In everyday use, Zoe browses the web with a screen reader and enjoys reading blogs. With TANGIBLESITE, she sets out to create her first blog page from scratch on her own.

TANGIBLESITE comprises a tangible baseboard and a set of shape–adjustable brackets that represent the webpage canvas and elements, along with a companion browser interface that supports content entry (via voice or keyboard) and provides audio feedback. The browser interface communicates with the tangible baseboard to track Zoe’s actions, detect design issues, save content, render a live page preview, and maintain an up–to–date representation of the evolving page.

Place elements and receive immediate state confirmation. Zoe wants to write an introduction to her hometown, with short text paragraphs and a few photos and videos. She begins by creating a title. Zoe selects a text bracket and places it at the top of the baseboard. Magnets embedded in both the baseboard and bracket provide snap–to–grid alignment. The system detects the bracket’s type, location, and size, announces a spoken summary (e.g., “Text

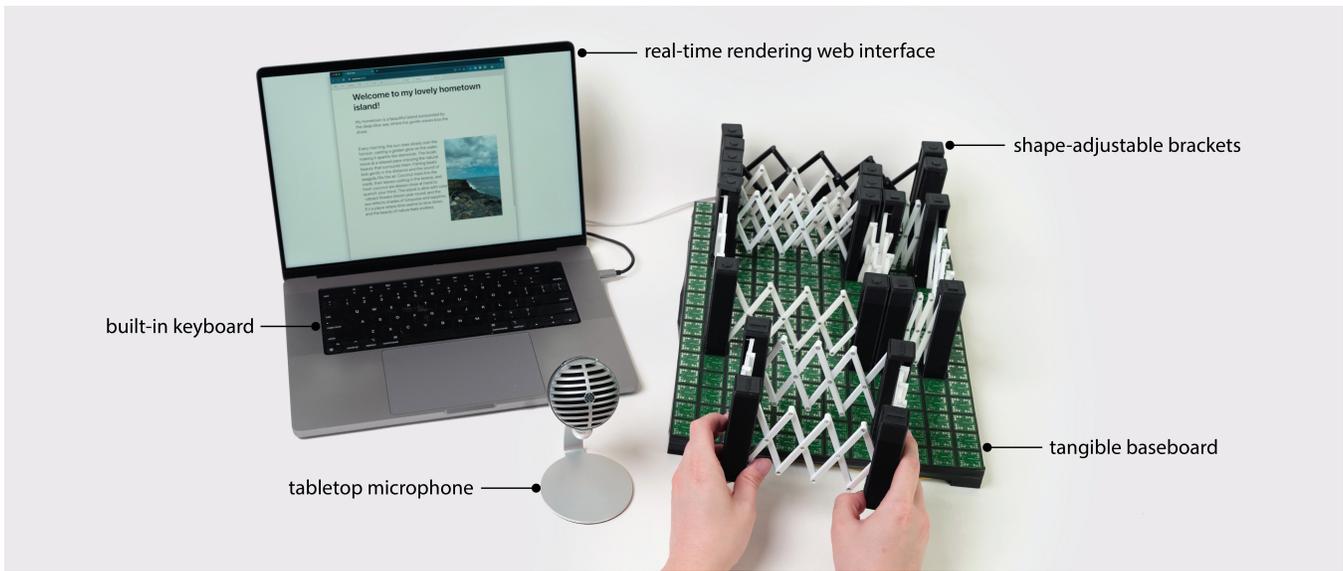


Figure 7: System overview. TANGIBLESITE supports accessible webpage authoring by coupling tangible layout manipulation with real-time multimodal feedback. Shape-adjustable brackets on a baseboard represent webpage elements and preserve structure across edits. Touch and audio feedback operate at both element and page levels, supporting inspection, verification, and overall layout understanding. A tabletop microphone enables voice commands, and a built-in keyboard supports text entry.

bracket detected, size 2 by 8, location at row 1 and column 3.”), and renders the corresponding container in the browser interface.

Enter content via voice or keyboard. To enter the title, Zoe touches the bracket and feels a vibration confirmation indicating that it is selected and ready for content entry. This confirmation clarifies when an element is selected for subsequent commands, reducing accidental edits or content entry on the wrong element. She then says the wake word followed by the “Title” command, dictates the title, and says “Stop” to end dictation. Later, when adding media, Zoe uses the browser interface with her screen reader and keyboard to select files for upload.

Iterate content and layout to resolve misfit. After Zoe enters the title, the system checks whether it fits inside the bracket and reports a mismatch (e.g., “Text exceeds bracket capacity. Consider shortening the text or expanding the bracket.”). Zoe enlarges the bracket by expanding it one column to the right. The system recognizes the resized bracket as the same element and confirms that the title now fits, preserving continuity as she iterates between content and layout. Zoe then adds a second text bracket for a short paragraph; after similar steps, the system announces element details and fit status to support iteration.

With the text in place, Zoe adds a photo. She picks up an image bracket, places it below the previous text bracket, and left-aligns it with the text. The system senses the bracket’s type, size, and location and provides a spoken summary. Zoe selects the bracket, issues the “Media” command, and a browser dialog opens for file selection. Using her screen reader and keyboard, she navigates to an image stored on her computer and uploads it. The system renders a live preview. If it detects a mismatch between the image orientation and the bracket dimensions, it provides an actionable recommendation

(e.g., “Image inserted; consider narrowing the bracket to reduce white-space.”). Zoe follows the suggestion by repositioning the image bracket with a narrower column width.

Review the evolving page and run checks on demand. Zoe repeats this workflow to add multiple text, image, and video brackets. When she forgets what an element contains, she touches the corresponding bracket and requests a brief element summary. The system reads the element’s type and content status, helping her reorient without reconstructing the page from memory.

Periodically, she issues the “Check” command, which reads aloud a structured layout summary, including the number and types of brackets on the baseboard and which grid regions are empty or occupied. The system flags issues (e.g., text overflow or media misalignment) and proposes fixes, reducing the cost of reorientation as the design grows. Zoe alternates between editing content and reshaping or repositioning brackets, allowing content and layout to co-evolve. After several content iterations and layout changes, Zoe is satisfied with her webpage and feels comfortable sharing it with her friends.

4.2 Interaction Design

The interaction design of TANGIBLESITE is grounded in the design opportunities distilled from our co-design sessions: the need for a persistent representation of the evolving layout, a way to distinguish user intentions during touch, and clear guidance when content and layout fall out of alignment. TANGIBLESITE integrates tangible, haptic, and speech-based modalities to support these needs.

Tangible interactions for maintaining spatial grounding (D1). During co-design, participants depended on stable spatial cues to rebuild their mental model of the page, especially once multiple

elements were present. TANGIBLESITE retains the familiar tactile grid and snap-in brackets introduced in prior work, but extends their function by linking every physical bracket to a persistent digital record of its type, size, and associated content. When a bracket is moved, resized, or temporarily lifted from the board, the system preserves its content and layout metadata, allowing users to restore or modify the element without losing track of what it represents. This integration of tactile structure with a persistent layout memory supports D1 by giving blind designers a reliable spatial anchor they can return to at any time, without having to reconstruct the entire page from memory.

Touch and haptic cues for clarifying system state (D2). Participants often touched brackets for different purposes: to review what an element contained, to confirm whether the system recognized their selection, or to prepare the next action. Since these intentions are indistinguishable from the system’s perspective without additional cues, TANGIBLESITE introduces context-aware haptic feedback that disambiguates these touch states. A brief vibration confirms that a bracket has been selected and that subsequent speech commands will be routed to it. A long press triggers a state readback that announces the bracket’s type, size, and any stored content or media. These immediate, touch-aligned cues reduce uncertainty about whether the system registered an action and help users maintain orientation as the layout changes. This design directly supports D2 by ensuring that tactile exploration and manipulation are always paired with clear system acknowledgment.

Audio guidance for understanding content–layout relationships (D3). Co-design participants did not want more description. Instead, they wanted information that helped them decide what to do next. To support this need, TANGIBLESITE provides audio feedback that explains how content and layout interact, paired with concrete adjustment options.

When text is entered, the system checks fit and reports the information exactly as implemented, for example: *“The current number*

of characters inside the bracket is 20. The maximum is 15. The recommended number is 10.” This allows users to determine whether to revise wording or resize the bracket.

When images or videos are inserted using the “Media” command, TANGIBLESITE opens the folder containing all the pre-prepared media files, enabling navigation with standard screen reader tools. After upload, the system evaluates the media–container relationship and verbalizes an actionable recommendation, such as *“Image inserted; one column remains empty on the left and right. Consider narrowing the bracket by one column on each side.”*

Speech commands are organized around common authoring tasks. Commands such as “Title”, “Text”, “Next line”, and “Stop” control content entry. The “Check” command retrieves a summary of the current webpage state: the system verbalizes the total number of each bracket type on the baseboard and calculates the percentage of whitespace not occupied by brackets. This summary highlights unresolved overflow, underflow, or spatial gaps without requiring users to sweep the board manually.

Together, these verbal cues turn audio feedback into a decision-making resource. Rather than passively narrating the layout, TANGIBLESITE explains fit, points out mismatches, and suggests next steps. This realizes D3 by giving blind designers the information and options they need to iterate with intention and resolve content–layout conflicts nonvisually.

Supplementary keyboard input for precise text refinement (D3). Although speech-based input is the primary method for entering and revising content, TANGIBLESITE also supports conventional keyboard interaction. This option gives blind designers greater control over fine-grained edits, such as selecting, deleting, or rearranging text. Many blind individuals already work efficiently with screen reader software and familiar shortcuts like “Ctrl + C” and “Ctrl + V,” so the keyboard serves as a natural extension of the system’s audio-guided content editing. This complements the actionable feedback in D3 by enabling text refinement that dictation alone may not provide.

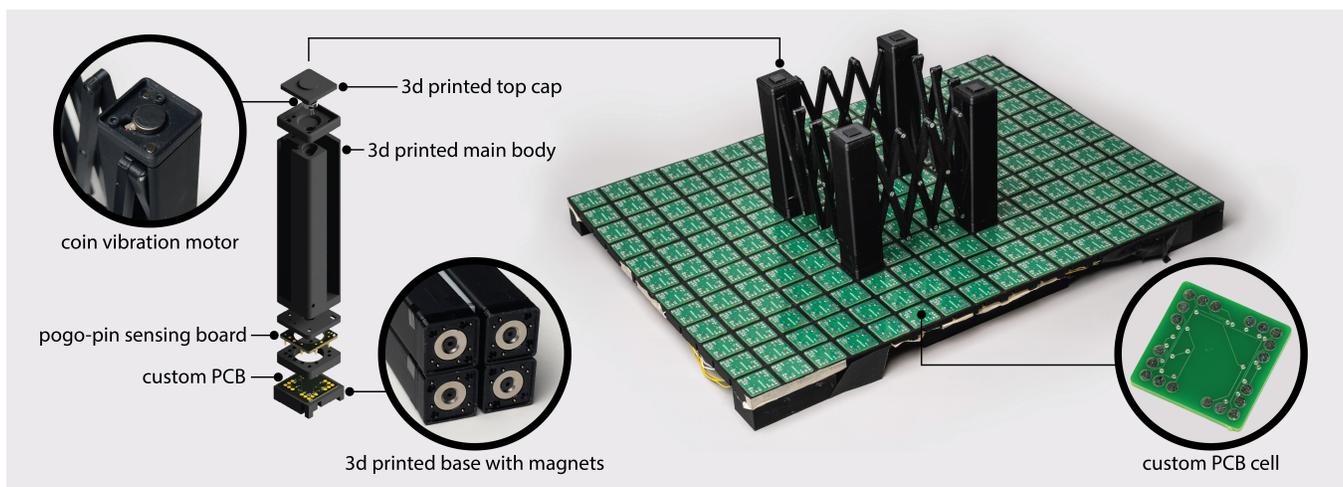


Figure 8: TANGIBLESITE hardware components. The baseboard detects bracket type, location, and size using custom PCB cells. An exploded view shows the modular pillar design with magnetic attachment, electrical sensing, and vibrotactile feedback.

4.3 System Implementation

4.3.1 Hardware. As shown in Figure 8, the baseboard and tangible brackets of our system are inspired by TangibleGrid [46], but incorporate several key design differences. The baseboard measures 305 mm × 406 mm and contains a grid of 192 cells. Each cell serves two functions by physically aligning brackets placed on top and electronically connecting to them for sensing and vibration. To support these functions, each cell in TANGIBLESITE has a layered structure, consisting of a 3D-printed holder at the base, a bolted countersunk ring magnet (15 mm diameter, 3 mm thickness), and a custom-designed PCB at the very top. The PCB contains 20 contact points arranged in central symmetry, together with the ring-shaped magnet, ensuring that brackets connect correctly to the baseboard regardless of orientation. The circuit detects bracket touch and activates vibration feedback when required.

The tangible bracket, shown in Figure 8, consists of four pillars connected by scissored linkages. While its appearance resembles that of TangibleGrid, the implementation differs substantially. Each pillar is designed as a multi-section structure to support both direct touch and localized vibration feedback. At the top, a 3D-printed tactile pattern made from conductive PLA filament enables capacitive touch sensing. Directly beneath the pattern, a vibration motor is embedded to provide vibrotactile feedback, with its pins routed through the pillar to a custom PCB concealed at the base. The PCB incorporates five spring-loaded pogo pins that interface with the physical baseboard, along with a surface-mount resistor that allows the TANGIBLESITE system to automatically identify the bracket type, compute its size, and register touch interactions. Finally, a ring magnet matching that of the baseboard is bolted to the very bottom of the pillar, flush with the pogo pins, to ensure secure physical alignment at all times.

The size of the bracket is adjustable, ranging from 50 mm × 50 mm to 300 mm × 300 mm. The maximum width of 300 mm allows the bracket to cover the entire width of the web canvas, which is useful when users want to design full-width elements, such as banners. The height of the bracket is fixed at 112 mm.

4.3.2 Software. Our software consists of three main components: (1) the front end, which renders the webpage and provides audio feedback; (2) the back end, which communicates with the TANGIBLESITE hardware to receive information about the brackets; and (3) the database, which stores the web layout design and content. The front end is built with the React framework and TypeScript. We utilize the SpeechRecognition Web Speech API for voice recognition and the SpeechSynthesis Web Speech API for audio feedback. The front end is supported by a back end server running on Express, a minimal web framework in Python. The server receives all web layout design information, such as bracket size and location, from the TANGIBLESITE hardware and stores it in the database.

5 USER EVALUATION

We conducted a formative user study to evaluate the effectiveness of TANGIBLESITE in enabling blind individuals to create and iterate on their own webpage.

5.1 Participants

We recruited a total of six participants, including five females and one male. Three took part in the co-design study, while the other three participants were newly recruited through a local NFB email group. Among participants who disclosed their age, ages ranged from 19 to 69 years ($M = 45.4$, $SD = 16.6$); one participant chose not to disclose their age. All participants self-reported being blind.

The newly recruited participants (P4–P6) brought varied levels of web design experience. P4 had no prior experience with web design. P6 had some familiarity with WordPress but was only comfortable editing text content. P5 had extensive web design experience and was familiar with website template platforms such as WordPress and Drupal. They hired someone to initially create their personal website and now maintain it using HTML, CSS, and PHP.

5.2 Procedure

The study had two sessions: a practice session and an evaluation session. In the practice session, participants familiarized themselves with TANGIBLESITE through guided tasks such as creating and deleting webpage elements and requesting audio feedback about individual elements or the overall page layout. Participants were also encouraged to freely explore and practice both layout design and content entry.

Once participants became familiar with the system, they proceeded to the evaluation session, where they were tasked with designing a webpage based on one of the following themes: a portfolio, personal blog, travel diary, or small business homepage. To accommodate time constraints, participants were encouraged to brainstorm initial design concepts and prepare content in advance of the study visit. If needed, we also offered the option to generate content using ChatGPT during the session. Upon completing the evaluation session, participants took part in a post-study interview, which included open-ended questions and Likert-scale ratings to gather feedback on their experience using the TANGIBLESITE.

The entire study lasted approximately two hours. All participants were compensated at a rate of \$35 per hour. The study was approved by the university's IRB.

5.3 Study Results

All six participants completed a webpage during the evaluation session (Figure 9). Likert ratings in Figure 10 indicate that participants were able to use TANGIBLESITE to generate content, design webpage layouts, and iteratively revise web elements. Most participants noted that the combination of interaction modalities helped them understand both the overall webpage structure and the details within individual elements. Participants expressed enthusiasm about the design possibilities enabled by TANGIBLESITE, noting that it allowed them to accomplish tasks they had rarely considered feasible. Below, we organize findings by how D1–D3 manifested in independent authoring.

Persistent spatial grounding supported diverse iterative workflows (D1). Participants used the physical layout and in-system content memory to manage complexity and revisit earlier decisions. Those who preferred sequential workflows (P3, P4) finalized elements one at a time, relying on the system's representation and feedback about both structure and content.

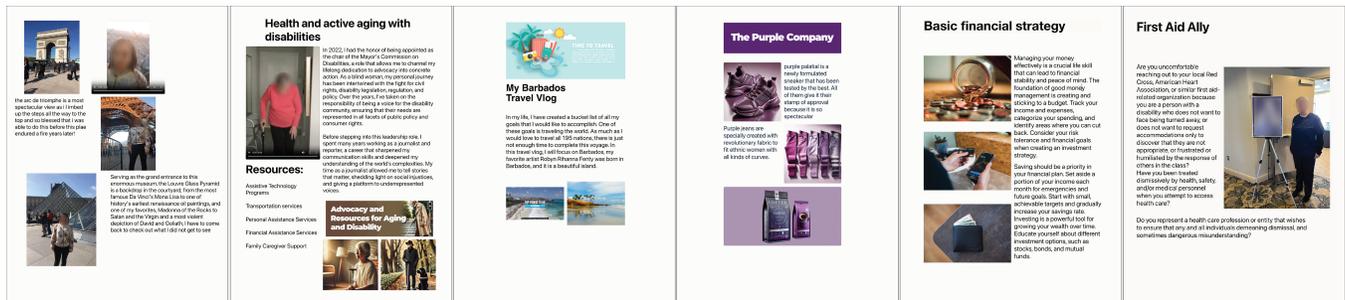


Figure 9: Six webpages designed by blind participants with TANGIBLESITE during the evaluation study.

“You know there are so many things to keep track of, and I forget things so I like to finalize each element before I move to the next.” – P3

Others leveraged D1 to support more flexible, global reasoning. P2 arranged all three image brackets before inserting photos, explaining that they wanted to understand spacing and size first. P4 experimented with relocating a text bracket to balance future photos across the page; when the system reported that the new location could not hold the original text, they restored the previous layout. Together, these behaviors show how persistent spatial grounding enabled exploration and reduced the risk of losing track of content or creating layout-content inconsistencies.

Touch-aligned confirmation increased confidence and reduced errors (D2). Participants used touch both to understand the layout and to verify selection and system status. Many described vibration feedback as essential for maintaining confidence during manipulation. P2, for example, emphasized that vibration provided immediate confirmation that the intended bracket was selected, increasing their sense of control during the design process. Participants also used touch-triggered readbacks to reorient themselves, reducing reliance on memory as layouts grew more complex. Overall, vibration confirmation and touch-triggered readbacks reduced

ambiguity and helped participants act on the intended element with confidence.

Actionable audio guidance enabled informed refinements to content and layout (D3). Real-time analysis of bracket content, spacing, and fit allowed participants to identify and correct issues they would otherwise miss. P1 repeatedly adjusted bracket sizes until all text fit, explicitly crediting the system’s feedback:

“It’s telling me how many characters belong in a specific box... So it’s giving me the necessary information that you need without overkill.” – P1

Similarly, when inserting images, participants responded to feedback about unused space or aspect-ratio mismatches by resizing brackets. For example, P5, who had web design experience, noted that combined tactile and audio feedback improved awareness of alignment and whitespace, which can be difficult to perceive non-visually (Figure 11).

Participants frequently used the “Check” command to receive an overview of the entire page, learn which regions were empty or occupied, and identify potential misalignment. Together with element-level recommendations, these page-level summaries supported both local fixes and global refinement.

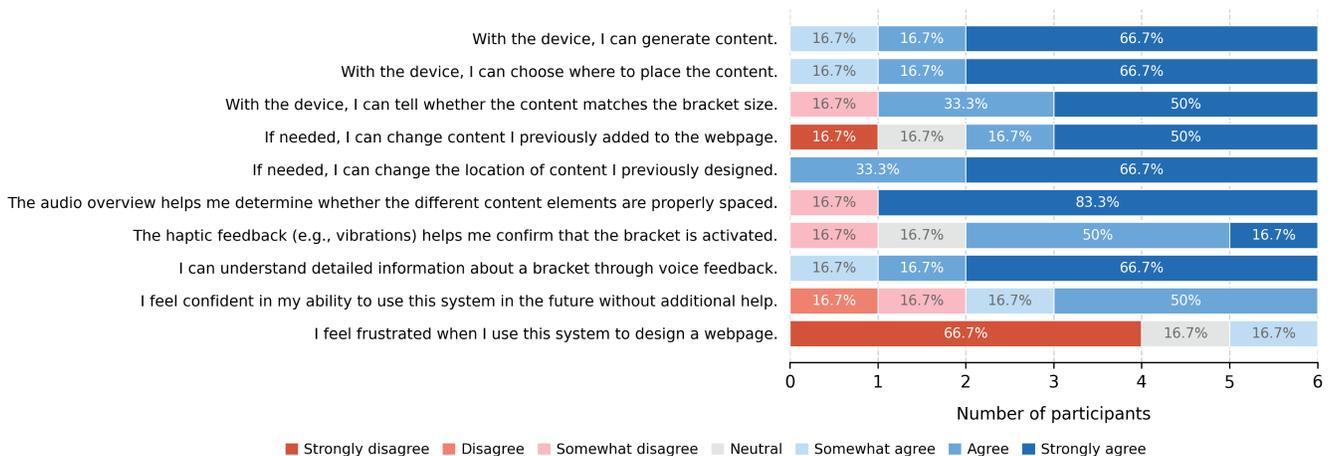


Figure 10: Participant self-reported ratings for ten Likert scale statements about web design experience with TANGIBLESITE.

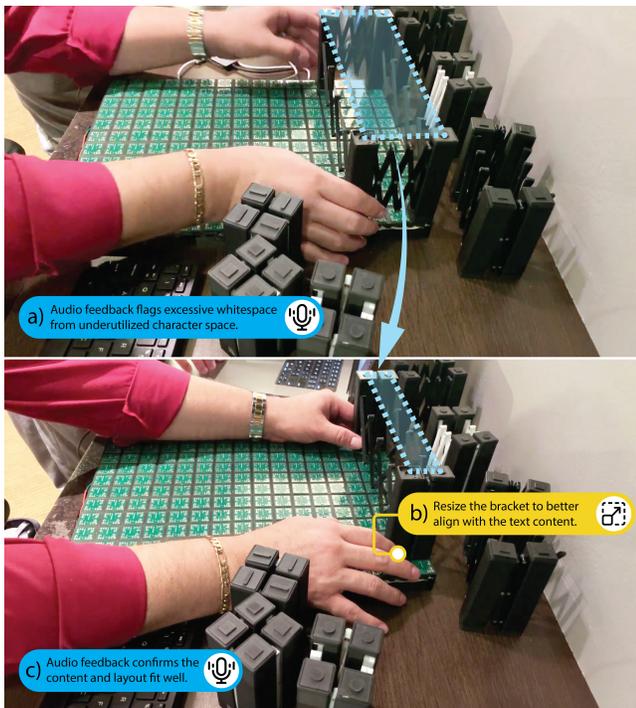


Figure 11: Iterative design refinements made during the evaluation session: The participant receives a) system-generated audio feedback detailing the current status of the text bracket. Based on this feedback, the participant decides to b) resize the bracket to better accommodate the content, continuing until c) the audio feedback confirms that both the content and layout fit appropriately.

6 DISCUSSIONS

6.1 Multimodal Balance: Supporting Flexible Design without Overload

Our co-design findings showed that blind participants prefer to approach webpage creation as a fluid process in which content and layout co-evolve. TANGIBLESITE supports this flexibility with persistent system memory (D1), immediate touch-aligned confirmation (D2), and actionable audio guidance (D3). However, this flexibility can still become fragile as design complexity increases. As multiple elements accumulated on the canvas, participants could still feel burdened by remembering prior decisions, monitoring content fit, and keeping track of spatial relationships.

These observations highlight an inherent tension. Providing full control over both content and layout increases creative freedom but also increases cognitive effort, especially for blind individuals who are new to web design. Several evaluation participants noted that they needed time to learn and recall commands and determine next steps, and relied heavily on system prompts during early use. This suggests that accessibility is not only a matter of adding features or additional modalities, but also of managing how much information and how many choices the system presents at any moment.

To address this tension, future versions of TANGIBLESITE could incorporate adjustable guidance levels. For novices, the system could reduce decision load through stepwise workflows (e.g., templates or guided build-up modes), limit available actions to those most relevant in context, and prioritize high-impact feedback such as overflow, misalignment, and spacing conflicts. The system could also apply constraints or offer auto-adjustments when layout limits are reached. Such scaffolding could help blind designers build an initial mental model of content–layout relationships while still supporting iterative refinement.

For experienced designers, TANGIBLESITE could provide richer but more compact summaries and more direct control, including intent-aware suggestions inferred from element type and structure (e.g., headings, bullet lists, banners, and image/video sections), to support complex designs without interrupting flow. Overall, adapting guidance to user expertise may preserve flexible content–layout iteration while reducing the likelihood of overload.

6.2 Opportunities for Adaptive and Semi-Automated Layout Support

The evaluation sessions showed that participants could reliably add, modify, and inspect individual elements with TANGIBLESITE. However, interactions involving many elements or large structural changes required significant manual effort. Participants often needed to lift and reposition multiple brackets, recheck content fit, and reestablish their mental model of the layout after each change. Even with strong physical affordances and system feedback, these actions can be taxing when repeated.

These observations suggest an opportunity for partial automation that preserves agency while reducing mechanical effort, for example through motorized actuation of the shape-adjustable brackets (or mechanisms beneath the baseboard that can reposition them). With explicit approval, the system could automate common operations such as aligning elements, snapping to the grid, redistributing spacing, or instantiating frequently used layout templates. Consistent with our earlier discussion of adaptive feedback, the system could offer multiple levels of assistance, ranging from lightweight alignment aids to higher-level template generation to accommodate different levels of web design experience.

Several participants expressed interest in recognizable layout patterns (e.g., blogs, portfolios, small business sites). Motorized action could allow the system to lay down a template directly on the baseboard, preserving the tactile grounding that supports D1 and D2 while reducing setup time. For novice designers, templates could provide an initial scaffold for understanding content–layout relationships and support focused content creation within a stable structure, potentially with optional auto-fit behaviors (e.g., resizing to avoid overflow). For more experienced designers, motorized reconfiguration could support complex structures that are difficult to realize nonvisually, such as floating navigation bars or pop-up windows, and could extend to responsive variants that adapt across device breakpoints.

Importantly, any automation should follow a preview-and-accept interaction pattern. The system can propose an adjustment based on goals or content constraints, but changes should be confirmed

through touch and verification of the resulting physical configuration. This approach may reduce repetitive effort while maintaining control and transparency in the design process.

6.3 Design Advanced Webpages with AI

While TANGIBLESITE currently supports basic design elements (text, image, video), modern webpages are typically far more complex, incorporating forms, animations, and dynamic layouts that are especially challenging for nonvisual design. Recent advances in generative AI make it possible to create more complex webpages through natural language prompts, suggesting a role for AI as a design assistant integrated throughout the TANGIBLESITE workflow.

Prior work demonstrates AI capabilities such as proposing text, generating images [26], and editing video clips [27], which could help blind designers move past blank-page setup. AI could also help translate high-level styling intent (e.g., theme, background color, contrast) into CSS/HTML, enabling appearance adjustments without writing code. Beyond styling, AI could synthesize interactive components or small programs, including simple games, and embed them within webpages to support richer interactions. Integrated into TANGIBLESITE, these capabilities could shift effort away from low-level implementation toward composition and higher-level design decisions.

6.4 Supporting Collaborative Design

While TANGIBLESITE is currently oriented toward independent use, we see its potential as a bridge between blind and sighted collaborators. Blind designers gain tactile and audio insight into structure and content, while sighted collaborators can view the visual rendering. Real-time synchronization between tangible and digital representations could enable parallel editing and shared reasoning across modalities.

Extending TANGIBLESITE for such collaborative use would require features that support shared understanding and co-creation. For example, real-time, fine-grained synchronization between the tangible and visual interfaces could allow both collaborators to work in parallel on the same design. Additional features, such as webpage templates and annotation tools, may also be needed to support a shared workspace. We see this as a promising direction for future development.

6.5 Limitations

We close by noting limitations that bound the scope of this work.

Limited layout expressiveness. TANGIBLESITE represents layout through fixed bounding boxes on a uniform grid, without margins, padding, or finer typographic hierarchy. This simplification limits the range of spatial relationships and styling decisions that can be authored with the current system.

Scope and accessibility of generated webpages. TANGIBLESITE outputs semantically structured HTML that reflects the tangible layout and authored content, but it does not currently support responsive behaviors, advanced styling, or full visual design control. The current pipeline also does not automatically generate accessibility metadata such as alt-text and labels.

Study duration and scope. Our evaluation involved single-session interactions with six blind participants. While the study

demonstrates feasibility and highlights promising patterns, longer-term use may reveal different strategies, breakdowns, or learning trajectories.

7 CONCLUSION

We present TANGIBLESITE, a multimodal website authoring tool that supports blind individuals in creating and iterating both content and layout within a single, integrated system. Informed by co-design sessions with blind participants, TANGIBLESITE maintains an up-to-date representation of the page's structure and authored content and provides multimodal feedback to support ongoing refinement. Our user study shows that TANGIBLESITE reduces barriers to web design and supports iterative workflows, enabling blind designers to exercise greater control and creativity.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under Grant No. 2229885 (NSF Institute for Trustworthy AI in Law and Society, TRAILS). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. An LLM service was used for grammar and text editing.

REFERENCES

- [1] Dustin Adams, Sri Kurniawan, Cynthia Herrera, Veronica Kang, and Natalie Friedman. 2016. Blind photographers and VizSnap: A long-term study. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*. 201–208.
- [2] Khaled Albusays, Stephanie Ludi, and Matt Huenerfauth. 2017. Interviews and Observation of Blind Software Developers at Work to Understand Code Navigation Challenges. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility (Baltimore, Maryland, USA) (ASSETS '17)*. Association for Computing Machinery, New York, NY, USA, 91–100. <https://doi.org/10.1145/3132525.3132550>
- [3] Amine Awada, Youssef Bou Issa, Joe Tekli, and Richard Chbeir. 2013. Evaluation of touch screen vibration accessibility for blind users. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*. 1–2.
- [4] Catherine M. Baker, Lauren R. Milne, and Richard E. Ladner. 2015. StructJumper: A Tool to Help Blind Programmers Navigate and Understand the Structure of Code. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (Seoul, Republic of Korea) (CHI '15)*. Association for Computing Machinery, New York, NY, USA, 3043–3052. <https://doi.org/10.1145/2702123.2702589>
- [5] Paul Blenkhorn and D Gareth Evans. 1998. Using speech and touch to enable blind people to access schematic diagrams. *Journal of Network and Computer Applications* 21, 1 (1998), 17–29.
- [6] Andy Borka. 2025. *Developer toolkit (DTK)*. https://github.com/ajborka/nvda_developer_toolkit Accessed on 04.7.2025.
- [7] Virginia Braun and Victoria Clarke. 2006. Using Thematic Analysis in Psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101.
- [8] Lorna M Brown, Stephen A Brewster, SA Ramloll, R Burton, and Beate Riedel. 2003. Design guidelines for audio presentation of graphs and tables. *International Conference on Auditory Display*.
- [9] Matt Calder, Robert F Cohen, Jessica Lanzoni, and Yun Xu. 2006. PLUMB: an interface for users who are blind to display, create, and modify graphs. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*. 263–264.
- [10] Tania Calle-Jimenez and Sergio Luján-Mora. 2016. Accessible Online Indoor Maps for Blind and Visually Impaired Users. In *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility (Reno, Nevada, USA) (ASSETS '16)*. Association for Computing Machinery, New York, NY, USA, 309–310. <https://doi.org/10.1145/2982142.2982201>
- [11] Cisco. 2017. *IntroToHTML5AndCSS*. <http://wiki.cucacat.org/index.php/Main/IntroToHTML5AndCSS> Accessed on 02.20.2024.
- [12] Katherine Adele Clark and Amy Pavel. 2025. Simultaneously Generating Multiple Mediums of Tactile Graphics. In *Proceedings of the 27th International ACM*

- SIGACCESS Conference on Computers and Accessibility (ASSETS '25). Association for Computing Machinery, New York, NY, USA, Article 166, 8 pages. <https://doi.org/10.1145/3663547.3759718>
- [13] CodeDesignAI. 2025. *AI Website Builder Build, Host & Export Websites*. <https://codedesign.ai/> Accessed on 08.20.2025.
- [14] Robert F Cohen, Arthur Meacham, and Joelle Skaff. 2006. Teaching graphs to visually impaired students using an active auditory interface. *ACM SIGCSE Bulletin* 38, 1 (2006), 279–282.
- [15] Kirk Andrew Crawford, Jennifer Posada, Yetunde Esther Okueso, Erin Higgins, Laura Lachin, and Foad Hamidi. 2024. Co-designing a 3D-Printed Tactile Campus Map With Blind and Low-Vision University Students. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility* (St. John's, NL, Canada) (ASSETS '24). Association for Computing Machinery, New York, NY, USA, Article 77, 6 pages. <https://doi.org/10.1145/3663548.3688537>
- [16] April R. Crockett and Gerald C. Gannod. 2020. Improving Understanding of Data Structures for the Blind with Tactile Media and a User-Centered Iterative Approach. In *2020 IEEE Frontiers in Education Conference (FIE)*, 1–8. <https://doi.org/10.1109/FIE44824.2020.9273978>
- [17] Franco Delogu, Massimiliano Palmiero, Stefano Federici, Catherine Plaisant, Haixia Zhao, and Olivetti Belardinelli. 2010. Non-visual exploration of geographic maps: does sonification help? *Disability and Rehabilitation: Assistive Technology* 5, 3 (2010), 164–174.
- [18] Md Ehtesham-Ul-Haque, Syed Mostofa Monsur, and Syed Masum Billah. 2022. Grid-Coding: An Accessible, Efficient, and Structured Coding Paradigm for Blind and Low-Vision Programmers. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) (UIST '22). Association for Computing Machinery, New York, NY, USA, Article 44, 21 pages. <https://doi.org/10.1145/3526113.3545620>
- [19] Google. 2024. *Use Google Sites with a screen reader*. <https://support.google.com/sites/answer/1637080?sjid=12764860936898643074-NA#zippy=%2Cvoiceover-screen-reader-shortcuts-macos> Accessed on 02.20.2024.
- [20] William Grussenmeyer and Elke Folmer. 2016. AudioDraw: user preferences in non-visual diagram drawing for touchscreens. In *Proceedings of the 13th International Web for All Conference*, 1–8.
- [21] Alex Hadwen-Bennett, Sue Sentance, and Cecily Morrison. 2018. Making programming accessible to learners with visual impairments: a literature review. *International Journal of Computer Science Education in Schools* 2, 2 (2018), 3–13.
- [22] Leona Holloway, Kim Marriott, and Matthew Butler. 2018. Accessible Maps for the Blind: Comparing 3D Printed Models with Tactile Graphics. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (CHI '18). Association for Computing Machinery, New York, NY, USA, 1–13. <https://doi.org/10.1145/3173574.3173772>
- [23] Leona Holloway, Kim Marriott, Matthew Butler, and Samuel Reinders. 2019. 3D Printed Maps and Icons for Inclusion: Testing in the Wild by People who are Blind or have Low Vision. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility* (Pittsburgh, PA, USA) (ASSETS '19). Association for Computing Machinery, New York, NY, USA, 183–195. <https://doi.org/10.1145/3308561.3353790>
- [24] Earl W Huff, Kwajo Boateng, Makayla Moster, Paige Rodeghero, and Julian Brinkley. 2020. Examining the work experience of programmers with visual impairments. In *2020 IEEE International Conference on Software Maintenance and Evolution (ICSM)*. IEEE, 707–711.
- [25] Mina Huh and Amy Pavel. 2024. DesignChecker: Visual Design Support for Blind and Low Vision Web Developers. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology* (Pittsburgh, PA, USA) (UIST '24). Association for Computing Machinery, New York, NY, USA, Article 142, 19 pages. <https://doi.org/10.1145/3654777.3676369>
- [26] Mina Huh, Yi-Hao Peng, and Amy Pavel. 2023. GenAssist: Making image generation accessible. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 1–17.
- [27] Mina Huh, Saelyne Yang, Yi-Hao Peng, Xiang'Anthony' Chen, Young-Ho Kim, and Amy Pavel. 2023. Avscript: Accessible video editing with audio-visual scripts. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1–17.
- [28] Deepak Jagdish, Rahul Sawhney, Mohit Gupta, and Shreyas Nangia. 2008. Sonic Grid: an auditory interface for the visually impaired to navigate GUI-based environments. In *Proceedings of the 13th International Conference on Intelligent User Interfaces* (Gran Canaria, Spain) (IUI '08). Association for Computing Machinery, New York, NY, USA, 337–340. <https://doi.org/10.1145/1378773.1378824>
- [29] Chandrika Jayant, Hanjie Ji, Samuel White, and Jeffrey P Bigham. 2011. Supporting blind photography. In *The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, 203–210.
- [30] Chandrika Jayant, Matt Renzelmann, Dana Wen, Satria Krisnandi, Richard Ladner, and Dan Comden. 2007. Automated tactile graphics translation: in the field. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*, 75–82.
- [31] Brigitte Jordan and Austin Henderson. 1995. Interaction Analysis: Foundations and Practice. *Journal of the Learning Sciences* 4, 1 (1995), 39–103.
- [32] Shaun K Kane, Jeffrey P Bigham, and Jacob O Wobbrock. 2008. Slide rule: making mobile touch screens accessible to blind people using multi-touch interaction techniques. In *Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility*, 73–80.
- [33] Shaun K Kane, Meredith Ringel Morris, Annuska Z Perkins, Daniel Wigdor, Richard E Ladner, and Jacob O Wobbrock. 2011. Access overlays: improving non-visual access to large touch screens for blind users. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 273–282.
- [34] Shaun K Kane, Meredith Ringel Morris, and Jacob O Wobbrock. 2013. Touchplates: low-cost tactile overlays for visually impaired touch screen users. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility*, 1–8.
- [35] Claire Kearney-Volpe. 2019. Web Development Training for Students That Are Blind. In *Proceedings of the 16th International Web for All Conference* (San Francisco, CA, USA) (W4A '19). Association for Computing Machinery, New York, NY, USA, Article 28, 2 pages. <https://doi.org/10.1145/3315002.3332434>
- [36] Claire Kearney-Volpe, Chancey Fleet, Keita Ohshiro, Veronica Alfaró Arias, Eric Hao Xu, and Amy Hurst. 2023. Tangible Progress: Tools, Techniques, and Impacts of Teaching Web Development to Screen Reader Users. *ACM Trans. Access. Comput.* 16, 1, Article 8 (mar 2023), 33 pages. <https://doi.org/10.1145/3585315>
- [37] Claire Kearney-Volpe and Amy Hurst. 2021. Accessible Web Development: Opportunities to Improve the Education and Practice of web Development with a Screen Reader. *ACM Trans. Access. Comput.* 14, 2, Article 8 (jul 2021), 32 pages. <https://doi.org/10.1145/3458024>
- [38] Claire Kearney-Volpe, Amy Hurst, and Scott Fitzgerald. 2019. Blind Web Development Training at Oysters and Pearls Technology Camp in Uganda. In *Proceedings of the 16th International Web for All Conference* (San Francisco, CA, USA) (W4A '19). Association for Computing Machinery, New York, NY, USA, Article 18, 10 pages. <https://doi.org/10.1145/3315002.3317562>
- [39] Andrea R Kennel. 1996. Audiograf: A diagram-reader for the blind. In *Proceedings of the second annual ACM conference on Assistive technologies*, 51–56.
- [40] Martin Kurze. 1996. TDraw: a computer-based tactile drawing tool for blind people. In *Proceedings of the second annual ACM conference on Assistive technologies*, 131–138.
- [41] Richard E Ladner, Melody Y Ivory, Rajesh Rao, Sheryl Burgstahler, Dan Comden, Sangeun Hahn, Matthew Renzelmann, Satria Krisnandi, Mahalakshmi Ramasamy, Beverly Slabosky, et al. 2005. Automating tactile graphics translation. In *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility*, 150–157.
- [42] Chunggi Lee, Sanghoon Kim, Dongyun Han, Hongjun Yang, Young-Woo Park, Bum Chul Kwon, and Sungahn Ko. 2020. GUIComp: A GUI design assistant with real-time, multi-faceted feedback. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, 1–13.
- [43] Jaewook Lee, Jaylin Herskovitz, Yi-Hao Peng, and Anhong Guo. 2022. ImageExplorer: Multi-layered touch exploration to encourage skepticism towards imperfect AI-generated image captions. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 1–15.
- [44] Seonghee Lee, Maho Kohga, Steve Landau, Sile O'Modhrain, and Hari Subramonyam. 2024. AltCanvas: a tile-based editor for visual content creation with generative AI for blind or visually impaired people. In *Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility*, 1–22.
- [45] Jingyi Li, Son Kim, Joshua A. Miele, Maneesh Agrawala, and Sean Follmer. 2019. Editing Spatial Layouts through Tactile Templates for People with Visual Impairments. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) (CHI '19). Association for Computing Machinery, New York, NY, USA, 1–11. <https://doi.org/10.1145/3290605.3300436>
- [46] Jiasheng Li, Zeyu Yan, Ebrima Haddy Jarjue, Ashrith Shetty, and Huaishu Peng. 2022. TangibleGrid: Tangible Web Layout Design for Blind Users. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology* (Bend, OR, USA) (UIST '22). Association for Computing Machinery, New York, NY, USA, Article 47, 12 pages. <https://doi.org/10.1145/3526113.3545627>
- [47] Yuwen Lu, Alan Leung, Amanda Swearngin, Jeffrey Nichols, and Titus Barik. 2025. Misty: Ui prototyping through interactive conceptual blending. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 1–17.
- [48] Sean Mealin and Emerson Murphy-Hill. 2012. An exploratory study of blind software developers. In *2012 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 71–74. <https://doi.org/10.1109/VLHCC.2012.6344485>
- [49] Abouakar Mountapmbeme, Obianuju Okafor, and Stephanie Ludi. 2022. Addressing Accessibility Barriers in Programming for People with Visual Impairments: A Literature Review. *ACM Trans. Access. Comput.* 15, 1, Article 7 (mar 2022), 26 pages. <https://doi.org/10.1145/3507469>
- [50] Joe Mullenbach, Craig Shultz, J Edward Colgate, and Anne Marie Piper. 2014. Exploring affective communication through variable-friction surface haptics. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 3963–3972.
- [51] Joe Mullenbach, Craig Shultz, Anne Marie Piper, Michael Peshkin, and J Edward Colgate. 2013. Surface haptic interactions with a TPad tablet. In *Adjunct Proceedings of the 26th Annual ACM Symposium on User Interface Software and*

- Technology. 7–8.
- [52] National Federation of the Blind. 2024. *National Federation of The Blind*. <https://nfb.org/> Accessed on 04.07.2025.
- [53] Hari Prasath Palani, G Bernard Giudice, and Nicholas A Giudice. 2018. Haptic information access using touchscreen devices: design guidelines for accurate perception of angular magnitude and line orientation. In *International Conference on Universal Access in Human-Computer Interaction*. Springer, 243–255.
- [54] Jamie Pauls. 2024. *Cisco Academy for the Vision Impaired: More Than an Education*. <https://www.afb.org/aw/17/11/15389> Accessed on 02.20.2024.
- [55] Yi-Hao Peng, Jason Wu, Jeffrey Bigham, and Amy Pavel. 2022. Diffscrber: Describing visual design changes to support mixed-ability collaborative presentation authoring. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–13.
- [56] Venkatesh Potluri, Liang He, Christine Chen, Jon E. Froehlich, and Jennifer Mankoff. 2019. A Multi-Modal Approach for Blind and Visually Impaired Developers to Edit Webpage Designs. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility* (Pittsburgh, PA, USA) (ASSETS '19). Association for Computing Machinery, New York, NY, USA, 612–614. <https://doi.org/10.1145/3308561.3354626>
- [57] Venkatesh Potluri, Priyan Vaithilingam, Suresh Iyengar, Y. Vidya, Manohar Swaminathan, and Gopal Srinivasa. 2018. CodeTalk: Improving Programming Environment Accessibility for Visually Impaired Developers. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) (CHI '18). Association for Computing Machinery, New York, NY, USA, 1–11. <https://doi.org/10.1145/3173574.3174192>
- [58] Emmanuel Schanzer, Sina Bahram, and Shirram Krishnamurthi. 2019. Accessible AST-Based Programming for Visually-Impaired Programmers. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education* (Minneapolis, MN, USA) (SIGCSE '19). Association for Computing Machinery, New York, NY, USA, 773–779. <https://doi.org/10.1145/3287324.3287499>
- [59] Ather Sharif and Babak Forouraghi. 2018. evoGraphs—A jQuery plugin to create web accessible graphs. In *2018 15th IEEE annual consumer communications & networking conference (CCNC)*. IEEE, 1–4.
- [60] Ather Sharif, Olivia H Wang, Alida T Muongchan, Katharina Reinecke, and Jacob O Wobbrock. 2022. VoxLens: Making online data visualizations accessible with an interactive JavaScript plug-in. In *Proceedings of the 2022 CHI conference on human factors in computing systems*. 1–19.
- [61] Ather Sharif, Andrew Mingwei Zhang, Anna Shih, Jacob O Wobbrock, and Katharina Reinecke. 2022. Understanding and improving information extraction from online geospatial data visualizations for screen-reader users. In *Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility*. 1–5.
- [62] Ashrith Shetty, Ebrima Jarjue, and Huaishu Peng. 2020. Tangible Web Layout Design for Blind and Visually Impaired People: An Initial Investigation. In *Adjunct Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology* (Virtual Event, USA) (UIST '20 Adjunct). Association for Computing Machinery, New York, NY, USA, 37–39. <https://doi.org/10.1145/3379350.3416178>
- [63] Xinyu Shi, Ziqi Zhou, Jing Wen Zhang, Ali Neshati, Anjul Kumar Tyagi, Ryan Rossi, Shunan Guo, Fan Du, and Jian Zhao. 2023. De-Stijl: Facilitating graphics design with interactive 2D color palette recommendation. In *Proceedings of the 2023 CHI conference on human factors in computing systems*. 1–19.
- [64] Chenglei Si, Yanzhe Zhang, Ryan Li, Zhengyuan Yang, Ruibo Liu, and Diyi Yang. 2024. Design2code: Benchmarking multimodal code generation for automated front-end engineering. *arXiv preprint arXiv:2403.03163* (2024).
- [65] Mathieu Simonnet, Anke M Brock, Antonio Serpa, Bernard Oriola, and Christophe Jouffrais. 2019. Comparing interaction techniques to help blind people explore maps on small tactile devices. *Multimodal Technologies and Interaction* 3, 2 (2019), 27.
- [66] Andreas Stefik, Roger Alexander, Robert Patterson, and Jonathan Brown. 2007. WAD: A Feasibility study using the Wicked Audio Debugger. In *15th IEEE International Conference on Program Comprehension (ICPC '07)*. 69–80. <https://doi.org/10.1109/ICPC.2007.42>
- [67] Andreas Stefik, Andrew Haywood, Shahzada Mansoor, Brock Dunda, and Daniel Garcia. 2009. SODBeans. In *2009 IEEE 17th International Conference on Program Comprehension*. 293–294. <https://doi.org/10.1109/ICPC.2009.5090064>
- [68] Jing Su, Alyssa Rosenzweig, Ashvin Goel, Eyal De Lara, and Khai N Truong. 2010. Timbremap: enabling the visually-impaired to use maps on touch-enabled devices. In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*. 17–26.
- [69] Amanda Swearngin, Mira Dontcheva, Wilmot Li, Joel Brandt, Morgan Dixon, and Amy J Ko. 2018. Rewire: interface design assistance from examples. In *Proceedings of the 2018 CHI conference on human factors in computing systems*. 1–12.
- [70] Steven Wall and Stephen Brewster. 2006. Feeling what you hear: tactile feedback for navigation of audio graphs. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*. 1123–1132.
- [71] Jason Wu, Eldon Schoop, Alan Leung, Titus Barik, Jeffrey P Bigham, and Jeffrey Nichols. 2024. Uicoder: Finetuning large language models to generate user interface code through automated feedback. *arXiv preprint arXiv:2406.07739* (2024).
- [72] Cheng Xu, Ali Israr, Ivan Poupyrev, Olivier Bau, and Chris Harrison. 2011. Tactile display for the visually impaired using TeslaTouch. In *CHI'11 Extended Abstracts on Human Factors in Computing Systems*. 317–322.
- [73] Tsubasa Yoshida, Kris M Kitani, Hideki Koike, Serge Belongie, and Kevin Schlei. 2011. EdgeSonic: image feature sonification for the visually impaired. In *Proceedings of the 2nd Augmented Human International Conference*. 1–4.
- [74] Lotus Zhang, Simon Sun, and Leah Findlater. 2023. Understanding Digital Content Creation Needs of Blind and Low Vision People. In *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility* (New York, NY, USA) (ASSETS '23). Association for Computing Machinery, New York, NY, USA, Article 8, 15 pages. <https://doi.org/10.1145/3597638.3608387>
- [75] Lotus Zhang, Zhuohao Zhang, Gina Clepper, Franklin Mingzhe Li, Patrick Carington, Jacob O Wobbrock, and Leah Findlater. 2025. VizXpress: Towards Expressive Visual Content by Blind Creators Through AI Support. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility*. 1–18.
- [76] Zhuohao Zhang, Ruiqi Chen, Mingyuan Zhong, and Jacob O Wobbrock. 2025. SlideAudit: A Dataset and Taxonomy for Automated Evaluation of Presentation Slides. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*. 1–23.
- [77] Zhuohao Zhang, Gene S-H Kim, and Jacob O. Wobbrock. 2023. Developing and Deploying a Real-World Solution for Accessible Slide Reading and Authoring for Blind Users. In *Proceedings of the 25th International ACM SIGACCESS Conference on Computers and Accessibility* (New York, NY, USA) (ASSETS '23). Association for Computing Machinery, New York, NY, USA, Article 47, 15 pages. <https://doi.org/10.1145/3597638.3608418>
- [78] Zhuohao Zhang, Haichang Li, Chun Meng Yu, Faraz Faruqi, Junan Xie, Gene SH Kim, Mingming Fan, Angus Forbes, Jacob O Wobbrock, Anhong Guo, et al. 2025. A11yShape: AI-Assisted 3-D Modeling for Blind and Low-Vision Programmers. In *Proceedings of the 27th International ACM SIGACCESS Conference on Computers and Accessibility*. 1–20.
- [79] Zhuohao (Jerry) Zhang and Jacob O. Wobbrock. 2023. A11yBoard: Making Digital Artboards Accessible to Blind and Low-Vision Users. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) (CHI '23). Association for Computing Machinery, New York, NY, USA, Article 55, 17 pages. <https://doi.org/10.1145/3544548.3580655>