EyeNav: Accessible Webpage Interaction and Testing using Eye-tracking and NLP

Juan Diego Yepes-Parra, Camilo Escobar-Velásquez
Universidad de los Andes, Colombia
{j.yepes, ca.escobar2434}@uniandes.edu.co

***Abstract*—In the field of human-computer interaction (HCI), alternative interaction methods are becoming increasingly popular and commercially available. From this opportunity came EyeNav, a system that combines eye-tracking and natural language processing (NLP) to enhance accessibility and enable automated test generation. The integration of these technologies for intuitive web interaction, using pointer control via gaze and natural language processing for interpreting user intentions, also presents a record-and-replay module for generating automated test scripts. Envisioned for users with motor disabilities, developers, usability testers, and general users interested in exploring novel multimodal web interactions. The ultimate goal is to demonstrate that this tool can be used not only as a possible assistive technology but also as an innovative approach to software testing. The tool is available publicly at https://thesoftwaredesignlab.github.io/EyeNav/, accompanied with a demonstration video https://thesoftwaredesignlab.github.io/EyeNav/video.html**

***Index Terms*—Eye-tracking; Automated Test Generation; Assistive Technology; Natural Language Processing; Web Applications; Accessibility.**

I. Introduction

Alternative interaction methods are becoming increasingly prevalent across modern computing systems. Devices such as smartphones and tablets [1], [2], wearables [3], and headsets [4]–[6] frequently incorporate novel input technologies, enabling more natural, inclusive, and adaptive user experiences [7], [8]. These emerging methods hold particular significance for accessibility applications, as they provide alternative means for users with physical limitations to interact with digital content [9].

Although eye-tracking has been the subject of research for many years [10], it is increasingly emerging as an input modality through its growing integration into everyday consumer devices. When implemented effectively, it enables precise, intuitive, and hands-free control [11].

On the contrary, natural language processing (NLP) is already well estabilised as a compelling input method, enabling users to interact with software through spoken commands. From voice-enabled code generation [12] to smart assistants, speech interfaces are widely researched and deployed. Still, these systems can become frustrating if user intent is misinterpreted or ignored, highlighting the importance of robust semantic parsing and contextual understanding [13], [14].

EyeNav is a tool developed to introduce a novel input method, integrating real-time eye-tracking with natural language interaction within web applications. This system facilitates interactions that are intuitive, accessible, and innovative.

Implemented as a Chrome extension, EyeNav allows users to interact with pages via gaze-based control and spoken commands. It also integrates a record-and-replay module to support automated testing workflows using the Gherkin standart.

Initially designed for users with motor impairments; it also holds potential for developers interested in hands-free browser control or rapid prototyping, usability professionals conducting accessibility evaluations in web environments, and general users exploring novel multimodal web interactions. By bridging assistive technologies and test automation, the prototype facilitates broader evaluation and integration of accessibility focused solutions within mainstream web contexts.

This paper outlines our approach overview, methodology, evaluation and discusses implementation and future work considerations for the system.

II. Related Work

A. Eye-tracking in HCI

Eye-tracking has long been used to study human cognition, especially in behavioral and psychological research. Early applications focused on observation rather than interaction. For example, Zelinskyi et al. developed a Chrome extension for collecting eye-tracking data for behavioral analysis [15], and Jacob and Karn emphasized its value in usability research for evaluating user interfaces [16].

As the field of HCI evolved, researchers began exploring eye-tracking as an input method. Gips et al. introduced an early eye-controlled system to support users with motor impairments [10], and more recent work has expanded into areas like immersive experiences [7] and AI-powered image editing [17]. Modern AR/VR devices such as the Meta Quest Pro, Pico 4 Pro, and Apple Vision Pro now integrate eye-tracking natively, with the latter relying entirely on eye and hand gestures for interaction [11].

Eye-tracking also plays a role in accessible technologies. Wang et al. developed GazePrompt, a reading aid for low-vision users that responds to gaze-based behavior by offering visual and auditory assistance [18]. Similarly, commercial devices like the Tobii Dynavox TD Pilot allow users to control iPad-based AAC systems using only their eyes [19].

2643-1572/25/$31.00 ©2025 IEEE
DOI 10.1109/ASE63991.2025.00374

B. Speech recognition NLP in HCI

NLP has similarly demonstrated significant potential in assistive contexts within HCI as an input method [20]. Conversational assistants represent a widely adopted interface paradigm enabled by NLP. Established systems such as Siri [21], Google Assistant [22], and Alexa [23] have become integrated deeply to many consumer devices. More recently, generative AI has facilitated the emergence of other assistants, including ChatGPT [24] and Gemini [25]. Notably, these assistants typically operate as discrete applications rather than as embedded control layers, meaning users must actively invoke them within specific contexts rather than relying on them for continuous, system-wide interaction.

In the context of accessibility, Girón-Bastidas et al. emphasize the effectiveness of NLP-based technologies in supporting users with hearing impairments [26]. Martínez et al. also developed a tool for simplifying online content, enabling understandability for people with cognitive disabilities [27]. Avalos et al. propose a context-based model that allows for browsing the web through voice. The system utilizes user utterances to command the system entirely; thereby enabling users with motor disabilities to engage with web content [28]. These efforts exemplify how NLP can be highly adaptable to many use-cases, and types of disabilities.

NLP also finds application across a variety of other domains. In educational contexts for instance, NLP-enabled assistive technologies have been shown to enhance learning by supporting more immersive and interactive experiences [29]. Additionally, conversational agents have been employed to facilitate a wide range of instructional interactions [14].

C. Multimodal interfaces

The integration of gaze data with speech input is an emerging focus in multimodal HCI research. Khan et al. developed a system that combines eye-tracking and voice commands for implicit interaction, dynamically attaching notes to points of interest based on gaze [30]. Lee et al. introduced GazePointAR, a wearable platform that leverages both eye-tracking and speech recognition to enable real-time image recognition and context-aware assistance [31]. Zhao et al. presented EyeSayCorrect, which fuses gaze and speech modalities to enhance the accuracy of speech-based text input [32]. In contrast, EyeNav is designed as an explicit, web-based interaction technique, distinguishing itself from prior implicit or application-specific approaches.

D. The Future of the Web

The increasing prevalence of eye-tracking in mainstream consumer devices presents new challenges and opportunities for web design. Panwar examines the evolution of web applications from their static origins to highly dynamic, complex systems. The study emphasizes the growing importance of multimodal interaction, predicting that future interfaces will increasingly combine touch, voice, text, and gesture to meet user expectations [33]. This underlines the need for web interfaces to adapt for supporting eye-based interaction, favoring larger and more visually distinct elements over traditional hyperlink-based controls [34].

E. Record-and-Replay Testing

Record-and-replay testing enables developers to capture user interactions during runtime and replay them for debugging, regression testing, or usability evaluation [35], [36]. In this context, this approach provides a practical means of reproducing complex interaction sequences in controlled environments. Consequently, record-and-replay testing is becoming an important component of usability studies and system validation in immersive and gaze-aware interfaces.

III. EyeNav

This section outlines the EyeNav system based on its workflow, as illustrated in Fig. 1. EyeNav is composed of 2 main modules: (i) a Chrome Extension sidebar and (ii) a backend module built in python. The Chrome extension sidebar, which functions as an adjacent webpage alongside the main content, presents the available verbal commands, allows to initiate a interaction session, and shows the interpreted verbal commands in realtime once a session has started (See Fig. 2). Once a session begins, the system orchestrates multiple components, including voice recognition, eye tracking, and interaction logging.

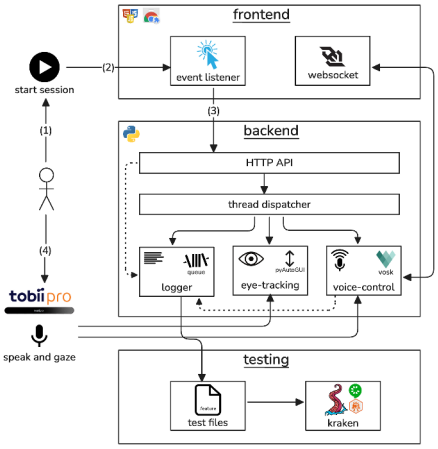

Fig. 1: Context diagram of the system.

User input is captured via an eye-tracker and microphone, while the underlying processing occurs in a backend service rather than on the frontend. Concurrently, user interactions, such as clicks, text input, and navigation events, are recorded by a logging module. These events are compiled into an executable test file, which can later be replayed using Gherkin-based test execution tools like Selenium [37] or Kraken [38].

A. High-Level Architecture

Figure 3 presents the complete component architecture of the system. The Chrome extension serves as the front-end interface, providing real-time textual feedback for recognized voice commands and capturing user interaction events. Click events are detected on the frontend and forwarded as HTTP

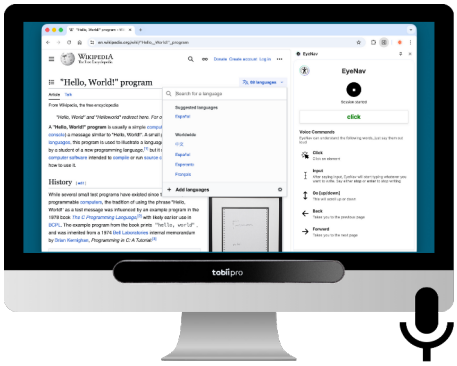

Fig. 2: A graphic of what the system looks like

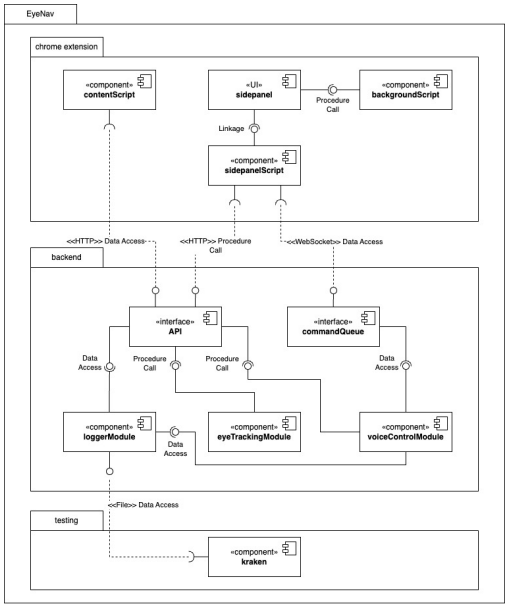

Fig. 3: Components diagram

POST requests to the backend to preserve contextual information, such as the associated HTML tag.

Eye tracking is powered by the Tobii Pro Nano, a single-camera dark/bright pupil system with corneal reflection and a typical latency of approximately 17 ms [39]. The gaze-driven pointer control module uses the `tobii-research` Python SDK to access real-time gaze data and interpolate cursor movement accordingly.

Voice commands are captured through a microphone and transcribed using the Vosk speech recognition engine [40], which operates entirely on-device for privacy and low latency. Recognized phrases are matched to predefined command templates and sent to the backend over WebSocket connections, enabling minimal response delay.

The backend integrates data from both the Tobii eye-tracker and the Vosk engine to interpret user intent, translate inputs into executable UI commands, and log all interactions. These are executed using Kraken, a behavior-driven testing framework built on WebdriverIO [41], supporting usability and regression testing even under dynamic UI conditions.

The system is organized into modular components, each adhering to a single-responsibility design principle. These include: (1) the gaze-driven pointer control, which enables real-time cursor positioning based on eye movement; (2) the NLP-based command parsing module, which interprets speech input captured by Vosk and maps it to specific UI actions among clicks, text entry, and scrolling; and (3) the record-and-replay test generation module, which logs user interactions in Gherkin syntax and compiles them into executable test scripts compatible with WebdriverIO for automated testing.

IV. EyeNav Capabilities

A. Voice Commands

The current implementation supports four voice commands: *Click*, *Input*, *Go (up/down)*, and *Navigate (back/forward)*. These commands were selected to reflect fundamental browser interactions typically performed via mouse and keyboard, providing a minimal yet functional command set suitable for a prototype. Each command corresponds to a distinct action: Click triggers a mouse click, Input captures and types dictated speech, Go scrolls the page vertically in the specified direction, and Navigate back or forward transitions between last and next pages in the browser history.

B. NLP in multiple languages

Accessibility also encompasses internationalization, and the system is designed to adapt to the user's preferred browser language. Currently, it supports both English and Spanish, with additional languages easily integrable by downloading the corresponding voice recognition model and translating the interface localization files.

C. Test Case generation

Interaction logging follows a structured hierarchy for element identification. For click events, the Chrome extension attaches a global event listener to the entire page, maintained dynamically via a `MutationObserver` to accommodate changes in the DOM. Only interactions with semantically clickable elements, including hyperlinks, buttons, and similar controls, are considered. When such an element is clicked, the system attempts to identify it using a prioritized attribute hierarchy: `href`, `id`, `className`, and, if necessary, an automatically computed `xPath`. This hierarchical strategy ensures that the most stable and descriptive selector available is used for referencing the element in the generated test script.

```
1  Feature: Replay of session on MM DD at HH:MM:SS [AM/PM]
2
3  @user1 @web
4  Scenario: User interacts with the web page named "Amazon.com.
       Spend less. Smile more."
5
6      Given I navigate to page "https://www.amazon.com/"
7      And I click on tag with id "twotabsearchtextbox"
8      And I input "nike black shoes"
9      And I click on tag with id "nav-search-submit-button"
10     And I scroll down
11     And I click on tag with xpath "/html[1]/body[1]/div[1]/div
           [1]/div[1]/div[1]/div[1]/span[1]/div[1]/div[9]/div[1]/
           div[1]/span[1]/div[1]/div[1]/div[1]/span[1]/a[1]"
```

The latter is an example of a generated test file. Each instruction is written in Gherkin syntax, promoting both human readability and maintainability. This format is particularly accessible to non-technical stakeholders while remaining fully compatible with the automated testing frameworks employed in the system.

D. Accessible Interaction

EyeNav supports keyboard navigation via the Tab key, enabling users to traverse interactive elements.

V. EyeNav Use Cases

Since each component operates independently, this flexibility allows the system to support a variety of use cases, each leveraging different capabilities of the tool.

A. Accessible Interaction Mechanism for Web Applications

Users with motor impairments or those seeking hands-free control can benefit from the gaze-based and voice-driven interface, offering an accessible and intuitive method of web navigation without the need for traditional input devices.

B. Record-and-Replay Testing Tool (A-TDD)

The logger module functions independently of the input modality, meaning it can be used even with conventional mouse and keyboard input. As such, EyeNav also serves as a lightweight, behavior-driven development (BDD) tool suitable for acceptance test-driven development (A-TDD) workflows. This enables teams to rapidly generate acceptance tests.

C. Support for Accessibility Evaluation Professionals

The system provides a valuable platform for consultants, QA engineers, and researchers conducting accessibility evaluations. Its support for multimodal input—via eye tracking, voice recognition, or a combination of both—enables flexible testing setups tailored to a wide range of users and scenarios, including those that simulate real-world constraints.

D. Intelligent Agents and Automation Scenarios

EyeNav's architecture allows for the integration of intelligent agents capable of interpreting and executing multimodal input. In envisioned scenarios, agents powered by NLP models could use EyeNav to autonomously interact, also using speech or other simulated modalities, with web interfaces. These interactions can also be recorded using the logger module, enabling automated usability assessments and supporting research in autonomous accessibility testing.

VI. Results

A. User Feedback

Qualitative interviews identified several key usability factors. Larger UI elements significantly improved gaze accuracy, making it easier for users to select targets with their eyes. Voice commands were most effective when they were short and distinct, reducing recognition errors. Environmental noise was found to negatively impact the reliability of speech recognition, suggesting the need for robust filtering or alternative input strategies. Users also indicated that visual indicators for "gaze hover targets" would enhance feedback and confidence during interaction. Additionally, simplified scroll commands were perceived as more usable and intuitive compared to earlier, more granular versions.

B. Accessibility Insights

Testers noted that this input method offers clear benefits for users with limited motor function. However, improvements in UI design (e.g., reachable icons) are needed for full accessibility compliance. Due to scope limitations, no participants with motor impairments were included, though future studies aim to address this.

VII. Discussion

The integration of eye-tracking and NLP proved effective for hands-free interaction in a browser context. Further work is needed to improve performance in varied environments (e.g., low-light, users with glasses, non-native accents). Eye-based clicking was responsive but may require calibration.

The testing module provided reproducible, human-readable test cases for interaction flows. While brittle on dynamic pages, these cases proved valuable for visual regression. Further validation with other similar tools is needed.

This architecture design allows for the combination of many of the purposes eye-tracking has already; we can analyze user behavior, identify usability bottlenecks, and validate system performance under varying conditions. This where subtle differences in gaze patterns can significantly influence interaction outcomes. Replay tools also facilitate comparative evaluations, allowing different interface versions or input modalities to be tested using identical interaction sessions.

VIII. Conclusions and Future Work

Planned improvements include enhancing support for users who wear glasses through prescription lens calibration techniques, and extending system compatibility to mobile and AR/VR platforms. Additional enhancements involve creating onboarding guides and integrating visual cues to facilitate learning gaze-based input, as well as including individuals with motor impairments in future user studies to better validate accessibility benefits, including quantitative performance metrics in the analysis. The system's test recorder is also being developed as a standalone tool to support broader usability testing efforts, including practical validation on the intelligent agents use case. Finally, there are plans to explore integration with immersive environments, such as the Apple Vision Pro and Meta Quest, to evaluate usability and responsiveness in mixed-reality contexts.

In summary, EyeNav demonstrates a novel fusion of eye-tracking and NLP to deliver an accessible, hands-free web-interaction paradigm while simultaneously generating executable, Gherkin-based test scripts. By orchestrating these modules, it provides developers and usability professionals an intuitive tool for rapid prototyping and automated testing.

REFERENCES

[1] Apple Newsroom. (2024, May) Apple announces new accessibility features, including eye tracking, music haptics, and vocal shortcuts. [Online]. Available: https://www.apple.com/newsroom/2025/05/apple-unveils-powerful-accessibility-features-coming-later-this-year/

[2] HONOR, "Honor magic6 pro," 2025. [Online]. Available: https://www.honor.com/co/phones/honor-magic6-pro/spec/

[3] Tobii AB, "Tobii glasses x: See. understand. improve." 2025. [Online]. Available: https://www.tobii.com/products/eye-trackers/wearables/tobii-glasses-x

[4] Apple Inc., "Apple vision pro," 2025. [Online]. Available: https://www.apple.com/apple-vision-pro/

[5] Sony Interactive Entertainment, "Playstation vr2 tech specs," 2025. [Online]. Available: https://www.playstation.com/en-us/ps-vr2/ps-vr2-tech-specs/

[6] HTC VIVE, "Vive pro 2 headset," 2025. [Online]. Available: https://www.vive.com/us/product/vive-pro2/overview/

[7] P. Dondi and M. Porta, "Gaze-based human–computer interaction for museums and exhibitions: technologies, applications and future perspectives," *Electronics*, vol. 12, no. 14, p. 3064, 2023.

[8] A. S. Fernandes, T. S. Murdison, and M. J. Proulx, "Leveling the playing field: A comparative reevaluation of unmodified eye tracking as an input and interaction modality for vr," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, 2023.

[9] Y.-H. Hsieh, M. Granlund, S. L. Odom, A.-W. Hwang, and H. Hemmingsson, "Increasing participation in computer activities using eye-gaze assistive technology for children with complex needs," *Disability and Rehabilitation: Assistive Technology*, vol. 19, no. 2, pp. 492–505, 2024.

[10] J. Gips and P. Olivieri, "Eagleeyes: An eye control system for persons with disabilities," in *The eleventh international conference on technology and persons with disabilities*, 1996, pp. 1–15.

[11] Z. Huang, G. Zhu, X. Duan, R. Wang, Y. Li, S. Zhang, and Z. Wang, "Measuring eye-tracking accuracy and its impact on usability in apple vision pro," *arXiv preprint arXiv:2406.00255*, 2024.

[12] Serenade Team, "Serenade - run tests with natural speech," 2025. [Online]. Available: https://serenade.ai

[13] N. Mozafari, W. H. Weiger, and M. Hammerschmidt, "The chatbot disclosure dilemma: Desirable and undesirable effects of disclosing the non-human identity of chatbots." in *ICIS*, 2020, pp. 1–18.

[14] J. Liu, "Chatgpt: Perspectives from human–computer interaction and psychology," *Frontiers in Artificial Intelligence*, vol. 7, p. 1418869, 2024.

[15] S. Zelinskyi and Y. Boyko, "Integrating session recording and eye-tracking: development and evaluation of a chrome extension for user behavior analysis," *Radioelectronic and Computer Systems*, vol. 2024, no. 3, pp. 38–54, 2024.

[16] R. J. Jacob and K. S. Karn, "Commentary on section 4. eye tracking in human-computer interaction and usability research: Ready to deliver the promises," *The mind's eye*, vol. 2, no. 3, pp. 573–605, 2003.

[17] R. Karlander and J. Wang, "Ai-assisted image manipulation with eye tracking," 2023.

[18] R. Wang, Z. Potter, Y. Ho, D. Killough, L. Zeng, S. Mondal, and Y. Zhao, "Gazeprompt: Enhancing low vision people's reading experience with gaze-aware augmentations," in *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–17.

[19] Tobii-A.B., "Td eye gaze pathway for td pilot: Instructions & resources," 2025.

[20] J. Song, "A review of the application of natural language processing in human-computer interaction," *Appl. Comput. Eng*, vol. 106, pp. 111–117, 2024.

[21] Apple Inc. (2011–2025) Siri. [Online]. Available: https://www.apple.com/siri/

[22] Google LLC. (2025) Google assistant. [Online]. Available: https://assistant.google.com/

[23] Amazon.com, Inc. (2025) Amazon alexa. [Online]. Available: https://developer.amazon.com/alexa

[24] OpenAI. (2022) Introducing chatgpt. [Online]. Available: https://openai.com/index/chatgpt/

[25] Google DeepMind. (2024–2025) Gemini. [Online]. Available: https://gemini.google/about/

[26] J. P. Girón Bastidas, O. J. Salcedo Parra, and M. J. Espitia R, "Natural language processing services in assistive technology," *International Journal of Mechanical Engineering and Technology*, vol. 10, no. 7, 2019.

[27] P. Martínez, L. Moreno, H. Ochoa, A. Ramos, and M. Pérez-Enríquez, "A tool suite for cognitive accessibility leveraging easy-to-read resources and simplification strategies," *CEUR-WS. org*, 2024.

[28] C. S. Avalos Montiel, J. G. Rodríguez García, S. Mendoza, and D. Decouchant, "Context-based model for browsing the web through voice," *Applied Sciences*, vol. 15, no. 6, p. 3400, 2025.

[29] G. Terzopoulos and M. Satratzemi, "Voice assistants and smart speakers in everyday life and in education," *Informatics in Education*, vol. 19, no. 3, 2020.

[30] A. A. Khan, J. Newn, J. Bailey, and E. Velloso, "Integrating gaze and speech for enabling implicit interactions," in *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022, pp. 1–14.

[31] J. Lee, J. Wang, E. Brown, L. Chu, S. S. Rodriguez, and J. E. Froehlich, "Gazepointar: A context-aware multimodal voice assistant for pronoun disambiguation in wearable augmented reality," in *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–20.

[32] M. Zhao, H. Huang, Z. Li, R. Liu, W. Cui, K. Toshniwal, A. Goel, A. Wang, X. Zhao, S. Rashidian *et al.*, "Eyesaycorrect: Eye gaze and voice based hands-free text correction for mobile devices," in *Proceedings of the 27th International Conference on Intelligent User Interfaces*, 2022, pp. 470–482.

[33] V. Panwar, "Web evolution to revolution: Navigating the future of web application development," *International Journal of Computer Trends and Technology*, vol. 72, 2024.

[34] Apple, "Wwdc24: Optimize for the spatial web," 2024. [Online]. Available: https://www.youtube.com/watch?v=5tjPBF2qoY4

[35] M. L. Vasquez, K. Moran, and D. Poshyvanyk, "Continuous, evolutionary and large-scale: A new perspective for automated mobile app testing," *arXiv preprint arXiv:1801.06267*, 2018.

[36] K. Moran, M. Linares-Vásquez, C. Bernal-Cárdenas, C. Vendome, and D. Poshyvanyk, "Automatically discovering, reporting and reproducing android application crashes," in *2016 IEEE international conference on software testing, verification and validation (icst)*. IEEE, 2016, pp. 33–44.

[37] B. García, C. D. Kloos, C. Alario-Hoyos, and M. Munoz-Organero, "Selenium-jupiter: A junit 5 extension for selenium webdriver," *arXiv preprint arXiv:2402.01480*, 2024.

[38] W. Ravelo-Méndez, C. Escobar-Velásquez, and M. Linares-Vásquez, "Kraken 2.0: A platform-agnostic and cross-device interaction testing tool," *Science of Computer Programming*, vol. 225, p. 102897, 2023.

[39] Tobii-AB, *Tobii Pro Nano: Enter the world of eye tracking research*, Tobii AB, Stockholm, Sweden, n.d. [Online]. Available: https://tobiipro.com

[40] Alpha Cephei, "Vosk speech recognition models," 2025, used English model (vosk-model-en-us-0.22) and Spanish model (vosk-model-small-es-0.42). [Online]. Available: https://alphacephei.com/vosk/models

[41] WebdriverIO Contributors, "Webdriverio: Next-gen browser and mobile automation test framework for node.js," 2025. [Online]. Available: https://webdriver.io