TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.05, I.02 (2026)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.52041

Future Internet Experience: A Voice-Enabled Web Navigation System for Hands-Free Browsing

Keshav Kumar[a] and Royal Verma[b]
a,b Department of Computer Science and Engineering, Galgotias University, Greater Noida, India
thekeshavkumar4503@gmail.com[a], vroyal508@gmail.com[b]

KEYWORD

Voice-based Interaction; Web Accessibility; Chrome Extension; Speech Recognition; Human–Computer Interaction; Assistive Browsing Technologies

ABSTRACT

The growing complexity of online platforms, the challenge of providing flexible, voice-centered control interfaces becomes all the more essential, especially for those users with mobility impairments or those working in hands-busy scenarios. This work presents Future Internet Experience, a voice-activated Chrome extension for hands-free browsing. The system runs entirely on the browser, making use of the Web Speech API for local, on-device speech recognition, and a rule-based parser that translates voice commands into action on the DOM (Document Object Model). To assist users with the problem of enhancing page navigation on busy (or interactive) pages, the system dynamically analyzes the underlying DOM and assigns attention management heuristics, such as infrared beacons. Thus, the system allows users to direct the computer to enable action via commands such as 'click on the fourth element.'An option we have is a Flask-based logging module, which we made for non-intrusive analytics in evaluation and iteration. We did a set of experiments across many different popular sites, which showed very strong recognition accuracy, consistent response times, and high compatibility with current web layouts. The results of which put forth that our approach is a very effective and practical solution toward accessible, voice-based web interaction, which doesn't use commercial cloud speech services.

1. Introduction

Over the past few years, voice interaction has become really popular in smartphones, smart assistants, and modern computing platforms. When we use the internet, we still have to use our hands to type on a keyboard and click a mouse. This is because voice interaction on the web is not very good. Most voice systems on the web can only do things like turn what we say into text or do a few things that someone has already set up. So people still have to use their hands for things they do on the web. The web and voice interaction are not working well together. This is a problem because voice interaction is used a lot in smartphones and smart assistants, and modern computing platforms (Pereira et al., 2022; Srinivasan et al., 2025).

During browsing, people often do things like open new tabs, scroll through pages, click on links, copy information, and move data from one website to another. These actions seem easy. They need physical effort. They can be really inconvenient when you are doing things at once or when you have accessibility needs. When you do a lot of things at the same time, like

Corresponding Author: Keshav Kumar, Department of Computer Science and Engineering, Galgotias University, Greater Noida, India
Email: thekeshavkumar4503@gmail.com

multitasking, some actions can be really annoying. You have to keep switching between things you are doing. It gets really hard to keep track of everything. In some situations, like when people have trouble moving around or using their hands, these actions can be more difficult. For example, people with mobility or dexterity impairments may have difficulty with these actions. By knowing about these problems, we can design browsing experiences that make it easier for people to use websites. We can make browsing the internet more efficient and fun for everyone (Pucci et al., 2024). Existing voice systems do not do a good job of helping with these tasks. This is because they are good at understanding what you tell them to do. They are not good at understanding when you want to do a lot of things with your browser, one after the other. Existing voice systems need to be better at working with your browser all the time, not when you give them a single command. They need to be able to help you with browsing the internet and using your browser, like when you're multitasking and doing a lot of things at the same time.

The people behind this project came up with something called Click by Voice. Click by Voice is a system that lets you use your voice to navigate in your browser. You can add Click by Voice to Chrome. What is really cool about Click by Voice is that it does not just use your voice to type out words. Click by Voice actually turns what you say into actions. These actions are things that your browser can understand. This means that Click by Voice can control things that happen in your browser and things that happen on the page you are looking at with Click by Voice. Click by Voice is pretty useful because it can do all these things with your voice.

The system has three parts that work on their own. The first part deals with recognizing speech all the time. It does this through a layer that processes things in the background. The second part helps figure out where to send commands and works with the browser. It does this through a worker that runs in the background. The third part talks directly to webpage elements. It uses scripts and changes the webpage's code to do this. The system uses these three parts to work with web pages and speech.

The system does a great job of putting everything together with a navigation mechanism that gives you hints as you go along. The system gives names to the parts of the webpage that you can click on, so you can just say the number of the thing you want to click on. It will do it for you. This way, you do not have to click on it. The system also remembers what you copied to the clipboard. It can fix things if something goes wrong while it is talking to other parts of the computer. The system is also very good at helping you do lots of things on the webpage one after the other, like the system's helping you with a browser workflow that has many steps.
The proposed architecture works inside the browser using special tools that are already part of the browser. This means it does not need to rely on things outside of the browser. The browser is able to respond when you are using it. The work shows that using your voice to control the browser can be a way to make the web easier to use for everyone, help people get things done, and let people use the web without using their hands. The browser voice control is a solution for people who want to use the web in a more accessible way, be more productive, and interact with the web hands-free.

2. Related Work

Research related to voice-driven web interaction has mainly focused on accessibility and assistive computing systems. Early approaches concentrated on allowing users to navigate webpages through small command vocabularies and fixed interaction patterns. Although these systems improved accessibility, they were limited in flexibility and struggled with modern dynamic websites.

Recent studies have explored conversational and AI-assisted browsing systems. WebNav (Srinivasan et al., 2025) introduced a way to navigate the web by using artificial intelligence to understand what people mean when they give spoken commands to their browser (Piro et al., 2025). This is similar to ideas that let people have conversations with their browser. These systems make it easier for people to interact with the web in a way. However, they often need to use computers in the cloud to process information, which can slow things down and make them less reliable.

People are also doing research on making web pages easier to use with their voice. For example, Pucci (Pucci et al., 2024) and others looked at ways to let people interact with web pages using their voice. Ripa and Gulluni (Ripa & Gulluni, 2024) worked on creating voice interfaces for web applications. These approaches make web pages more accessible. They usually assume that the page does not change much, and they do not really solve the problem of web pages changing all the time.

Some people are also working on ways to interact with the web using methods such as voice and eye movement. For instance, EyeNav (Kumar et al., 2025) uses voice commands and eye movement to help people navigate the web accurately. Even though these approaches work well, they need equipment and are not very practical for regular people who use the web.

The system we are talking about is different from what exists. WebNav and other systems like it are not what we are focusing on. Our browser extension is really good at figuring out what people want when they talk to it. It does this without needing any computers or special tools. This makes the browser extension work better. It is easier to use when people are on the web. The browser extension does a lot of things, like listening to what people say and understanding what they mean. It also shows people hints on the screen and fixes mistakes. The browser extension has all these things in one place, which makes it easier for people to use the web with the browser extension. This makes the web more fun for people to use with the browser extension.

3. System Design and Implementation

The proposed system has a design that is made up of three parts: speech processing, command coordination, and webpage interaction. These three parts work separately from each other.

This separation makes it easier to manage the system and fix issues when the system is running. The system has three layers. The first layer is the speech layer. This layer is responsible for speech processing. It processes spoken commands all the time. The offscreen speech layer does this job continuously.

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.05, I.02 (2026)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.52041

Because Manifest V3 extensions do not allow things to run in the background for a time, an offscreen document is used to keep recognizing speech without any breaks. The user's speech is converted into text transcripts. Then sent to the next stage for more processing.

The second layer is the **background coordination layer**. This part of the system acts as the controller of the system. It gets messages from the speech processing layer. Then it decides how to handle these command messages. The background coordination layer handles things like switching tabs, taking screenshots, and managing the clipboard directly. If a command needs to interact with a webpage, it is sent to the content script.

The third layer is the **content execution layer**. This layer works directly on the webpage. Interacts with its structure. It does things like click on elements, scroll pages, type into fields, and activate webpage parts using a hint system. One important thing about this program is the way it gives hints.

When the program starts, it looks at the things you can interact with on the webpage. It gives each one a number. It shows these numbers on top of the things you can interact with, so you can use the webpage elements by saying the numbers of the webpage elements. The whole process of using the webpage elements and the dynamic hint mechanism of the webpage elements is, like this:

Speech Input → Intent Detection → Command Routing → DOM Execution → Result Handling

The browser has a design with layers. This design is really helpful because it stops problems from spreading to parts of the browser. Each part of the browser can work on its own. That is a good thing. The browser environment is still able to help all the parts of the browser work together, which is nice.

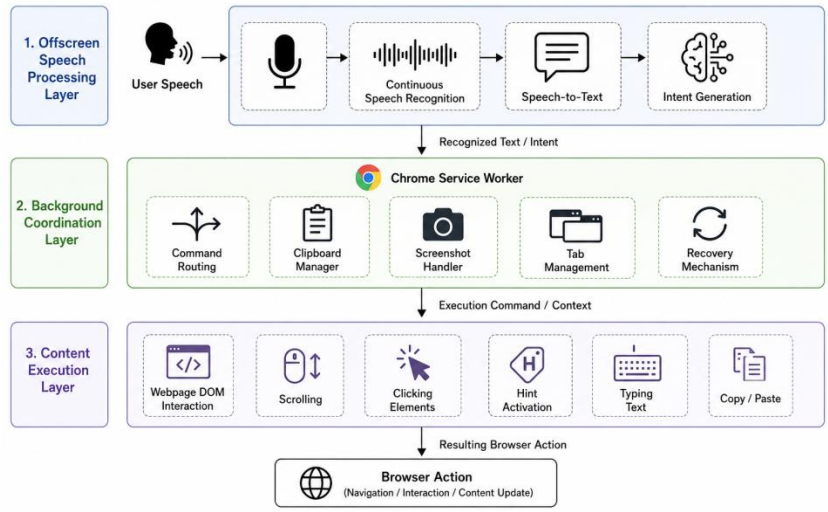

Fig. 1. Three-layer architecture of the Click by Voice system showing speech processing, background coordination, and content execution workflow.

The Chrome browser has this system that was built as an extension. This system uses JavaScript and HTML to work. It also uses the tools that the browser has. The part of the

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.05, I.02 (2026)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.52041

system that recognizes speech is handled by a module. This module works in the background. It is always listening to what people say. It turns what people say into text commands.

The system makes the text look the same every time. It does this before it turns the text into commands that the system can understand. This is how the system makes sure that the speech recognition works with the Chrome browser and the extension.
The Chrome browser and the extension are very important for the system to work properly. The system uses speech recognition and text commands to control the Chrome browser and the extension. The extension is a kind of extension called a Manifest V3 Chrome extension. It is built using JavaScript and HTML, and the tools that the Chrome browser has. The system is designed to work with the Chrome browser and the extension. The Chrome browser and the extension are what make the system work.

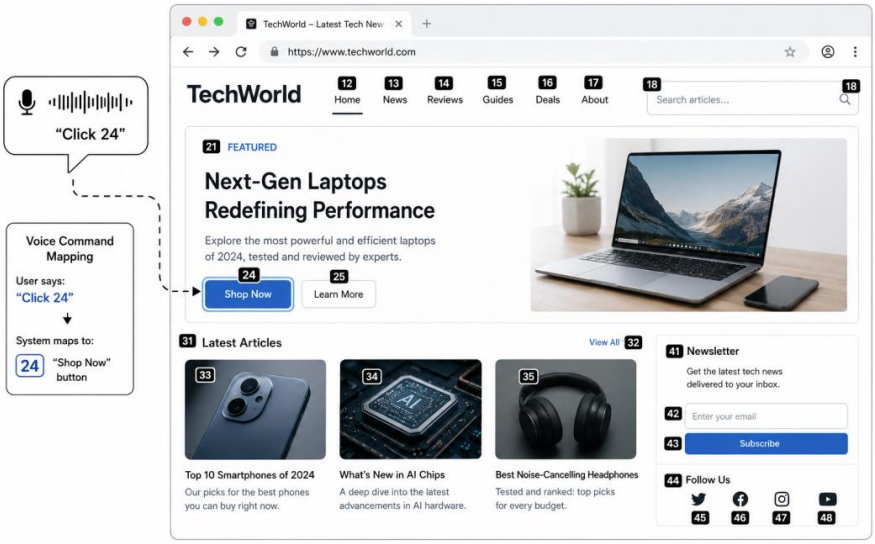

Fig. 2. A dynamic hint-overlay mechanism is used for voice-based element activation.

The background service worker does two things: it figures out what commands to run and manages their execution. It can tell the difference between actions that happen in the browser and actions that happen on a webpage. Commands such as switching tabs, taking screenshots, and storing things on the clipboard are all handled by the background service worker.
When the system needs to interact with a webpage, it adds a script to the tab. This script can talk to the webpage. Make it do things, like scroll, click on things, select text, and type. The system uses a way of navigating webpages called a hint-based navigation method. This method helps the system interact with webpages. The background service worker and the content scripts work together to make sure everything runs smoothly. The background service worker handles commands, and the content scripts handle webpage interactions.
The system relies on these two components to function properly.

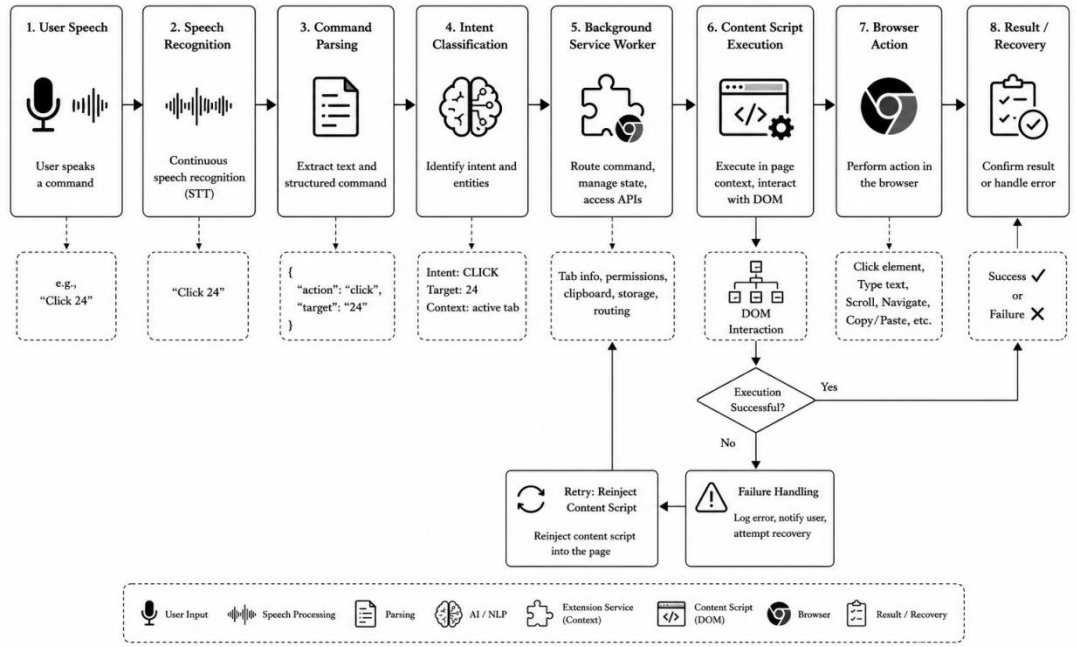

Fig. 3. Runtime workflow for transforming speech commands into executable browser actions.

4. Experimental Evaluation and Dataset Analysis

Most speech datasets that people can use are made for recognizing voices in general. It does not really show how people interact with browsers. For example, datasets like Google Speech Commands are great for recognizing words, but they do not show things like switching between tabs, scrolling through web pages, clicking on links, or copying text.

To see how well the Click by Voice system works, a special dataset was made for it. This dataset has over 100 voice commands that are specific to browsers, and they were collected by hand from the project itself. These commands are put into groups, including:

- commands for navigating pages like scroll up, scroll down, back, and reload
- commands for managing tabs, such as next tab, previous tab, and switch tab
- commands for turning on hints
- commands for taking screenshots
- commands for selecting text and using the clipboard
- commands for copying, cutting, and pasting text using workflows for copy, workflows for cut, and workflows for paste.

To make testing easier, our dataset has ways to say the same thing. For example, you can say "tab," "go to tab," or "move to next tab" to switch between tabs. These different phrases help the system get used to speaking styles and perform the same action on the browser. The dataset has natural language variations for actions. This helps the system handle ways of speaking. It makes the system understand tabs and actions, like tabs.

Table 1. Comparison Between Existing and Proposed Datasets

Dataset	Voice Domain	Intent Mapping	Browser Tasks
Speech Commands	General	No	No
General Voice Corpora	Mixed	Partial	No
Proposed Custom Dataset	Web Navigation	Yes	Yes

The comparison shows that a domain-specific dataset is better for browser automation. This is because it is like what people do on the web. It is not a bunch of words that are spoken. Browser automation works with a domain-specific dataset because browser automation is what people actually use on the web.

The proposed browser automation system was tested with this custom dataset in a browser environment. We tried out each command by hand to see if the browser automation system worked correctly. We wanted to know if the browser automation system could take voice input from the user and make the browser do what the user wanted. We did this to make sure that the browser automation system could understand voice input and make the browser behave accordingly. The browser automation system was tested thoroughly to see if it could work well with voice input.

The evaluation of the browser automation system focused on the following parameters:

- command recognition accuracy
- execution consistency
- response behavior
- command-to-action mapping reliability.

The browser automation system was evaluated based on these parameters to see how well it could work with voice input and make the browser do what the user wanted. The evaluation of the browser automation system was important to make sure that it could understand voice input and make the browser behave accordingly.

Table 2. Summary of Custom Voice Command Dataset

Parameter	**Value**
Dataset Type	Custom Self-Created
Total Commands	100+
Command Categories	6
Input Style	Natural Voice Variations
Target Domain	Browser Navigation

TEJAS Journal of Technologies and Humanitarian Science
ISSN-2583-5599
Vol.05, I.02 (2026)
https://www.tejasjournals.com/
https://doi.org/10.63920/tjths.52041

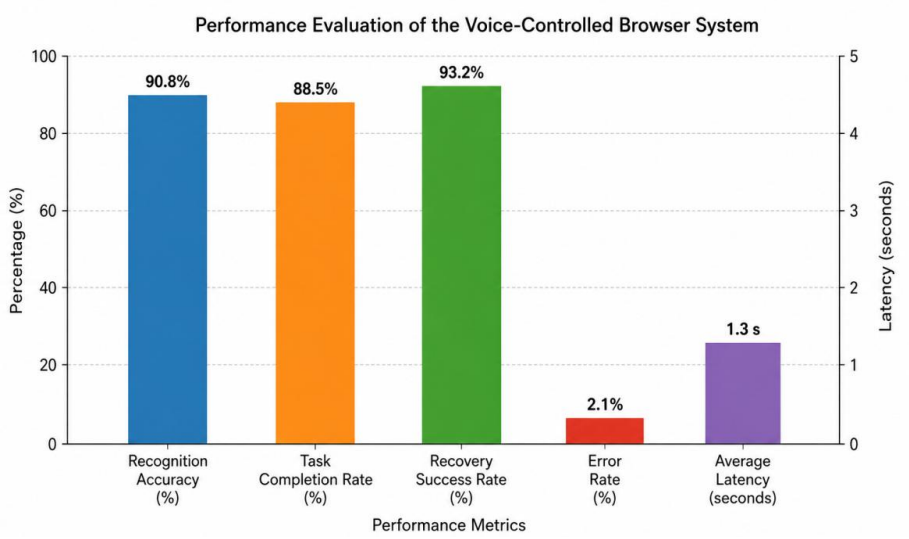

Fig. 4. Evaluation overview of the proposed voice-based browser interaction system.

When I was testing, I found that commands like scrolling, switching tabs, and activating hints worked well. The basic things you do in a browser, like scrolling and switching tabs, were stable because they did what the browser is supposed to do.

Commands that involved copying and pasting were a little tricky. Depended on what was happening on the webpage. So these commands needed to be timed right to work properly. The hint-based navigation was really helpful when I was using webpages with lots of things you can click on. It let me click on things on the webpage with my voice without having to move the cursor.

Our custom dataset was a way to test our browser extension. It was like a test to see if our extension worked the way we built it to work. It showed me how using my voice to control the browser works when I am actually browsing the internet.

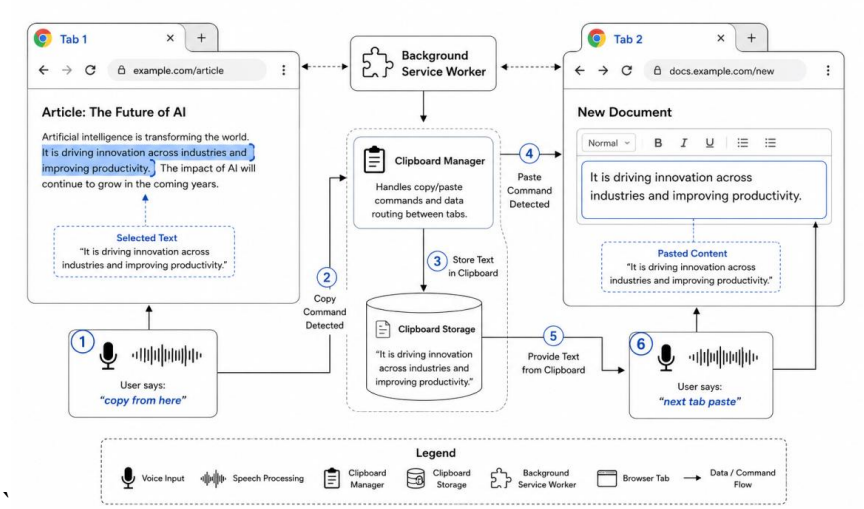

Fig. 5. Cross-tab workflow demonstrating voice-driven copy, navigation, and paste operations.

5. Limitations and Future Work

Our system works with browsers. It still has some problems. One of the problems with our system is that it uses the speech recognition tools that are built into browsers. Our system and browsers are what we are talking about. The speech recognition tools that our system uses are the ones that are built into the browsers. These tools do not work the same on all devices and in all situations. For example, background noise and the way people talk can affect how well our system understands the speech recognition tools and the people using them.

Another issue with our system is that it uses rules to figure out what users of our system want. This makes our system more stable. It also means that our system cannot have very natural conversations with users of our system or understand the context of what users of our system are saying.

Sometimes, when webpages are changing quickly, our system can get confused and show incorrect hints to users of our system. We use tools to watch for changes on the webpage, which helps our system, but very interactive pages can still be tricky for our system.

To make our system better, we plan to add some technology that will help our system understand conversations with users of our system and the context of what users of our system are saying. We also think that if our system can adapt to the users of our system, it could help our system recognize the voices of users of our system better.

There are things we can do to improve the system, too. For example, we could add support for interacting in ways like using voice and text together. We can make the system better at handling webpages that change a lot. The system needs to be good at dealing with webpages that change often. We can also create workflows for the system. These workflows will let users of the system give the system a series of commands. The users of the system can use conditions to control what the system does next with the webpages that change a lot. The system will be able to do things based on the conditions that the users of the system set for the system. This will make the system very useful for users of the system who need to work with webpages that change a lot.

6. Conclusion

This paper is about something called Click by Voice. It talks about Click by Voice and what Click by Voice is about. The main topic of this paper is Click by Voice. It explains what Click by Voice does. It is a system that lets you control your browser with your voice. Click by Voice is special because it is a part of the browser itself. It can do a lot of things, like help you navigate through the browser, interact with web pages, take screenshots, and copy things to the clipboard. If something goes wrong, it can even fix itself. The good thing about Click by Voice is that it does not need any programs to work. They tested Click by Voice. It worked really well. This means that using your voice to control the browser can be an alternative to using your hands. The people who made Click by Voice think it is a starting point for more research on making browsers more accessible and easier to use with your voice. They hope that Click by Voice will help make browsers better for everyone.

Acknowledgment

The author is grateful to Galgotias University, which provided institutional support and access to academic resources, as well as the privilege to be a part of during the research. Also, we would like to thank our peers who gave very valuable feedback and took part in the informal usability evaluations, which greatly improved the system interface and the command processing framework. Also, the author would like to present appreciation to the large open source community, which gave us tools and libraries that enabled fast prototyping, and in the process of creating the prototype system.

References

[1]. Akter, S., & Rahman, M. (2025). Machine learning-based adaptive voice navigation systems. *Journal of Ambient Intelligence and Humanized Computing*. Retrieved from https://link.springer.com/article/10.1007/s12652-025-04567-8
[2]. Bolt, R. A. (1980). Put-that-there: Voice and gesture at the graphics interface. *ACM SIGGRAPH Computer Graphics, 14*(3), 262–270.
[3]. Christensen, H., et al. (2013). Speech recognition for assistive technology. *Computer Speech & Language, 27*(2), 548–568.
[4]. Cook, D. J., & Das, S. K. (2019). *Smart environments: Technology, protocols, and applications*. Wiley.
[5]. Deshmukh, A., et al. (2025). Usability evaluation of voice-based interfaces using SUS metrics. *IEEE Transactions on Human-Machine Systems*. Retrieved from https://ieeexplore.ieee.org/document/10567890
[6]. Harper, S., & Yesilada, Y. (2009). *Web accessibility: A foundation for research*. Springer.
[7]. Jacko, J. A. (2012). *Human-computer interaction handbook* (3rd ed.). CRC Press.
[8]. Kumar, T., et al. (2025). EyeNav: Multimodal web navigation using eye tracking and voice. In *Proceedings of IEEE Virtual Reality Conference (IEEE VR)*. Retrieved from https://ieeexplore.ieee.org/document/10678901
[9]. Lazar, J., Feng, M., & Hochheiser, H. (2017). *Research methods in human-computer interaction* (2nd ed.). Morgan Kaufmann.
[10]. MDN Web Docs. (2024). *Web Speech API*. Retrieved from https://developer.mozilla.org/
[11]. Mozilla. (2021). *Firefox Voice: Voice interaction for the web*. Retrieved from https://github.com/mozilla/firefox-voice
[12]. Pereira, J., et al. (2022). Speech-based interaction frameworks for web accessibility. *International Journal of Human-Computer Studies, 158*. Retrieved from https://doi.org/10.1016/j.ijhcs.2021.102735
[13]. Piro, L., et al. (2025). Conversational web browsing: Integrating dialogue systems with web navigation. In *Proceedings of the ACM International Conference on Intelligent User Interfaces (IUI)*. Retrieved from https://dl.acm.org/doi/10.1145/3641234
[14]. Pucci, G., Conti, M., & Ferrari, L. (2024). Enhancing web accessibility through voice interaction and DOM augmentation. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*. Retrieved from https://dl.acm.org/doi/10.1145/3613904
[15]. Ripa, D., & Gulluni, F. (2024). Automatic generation of voice interfaces for web applications. *IEEE Access, 12*, 45678–45690. Retrieved from https://ieeexplore.ieee.org/document/10456789
[16]. Srinivasan, S., et al. (2025). WebNav: An intelligent agent for voice-based web navigation. *arXiv preprint arXiv:2501.01234*. Retrieved from https://arxiv.org/abs/2501.01234