iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** XI **Month of publication:** November 2025

DOI: https://doi.org/10.22214/ijraset.2025.75958

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

Voice Based Web Applications

Er. Harjasdeep Singh[1], Shalini Singh[2], Simranjeet Kaur[3]
[1]*Associate Professor, Department of CSE, MIMIT Malout, Malout, Punjab, India*
[2,3]*B.Tech, CSE, Department of CSE, MIMIT Malout, Malout, Punjab, India*

Abstract: ***Voice-based interfaces have significantly reshaped human–computer interaction by enabling natural, hands-free communication. While commercial systems such as Amazon Alexa, Google Assistant, and Apple Siri showcase the power of Automated Speech Recognition (ASR), their integration into modern web applications remains limited. This study exam- ines methods for embedding voice recognition within web ar- chitectures using frameworks such as the Web Speech API and JavaScript libraries. It further evaluates usability, acces- sibility, and technical challenges—including background noise, multilingual limitations, privacy concerns, and inconsistent user experiences. By combining technical analysis with accessibility-driven design principles, this research bridges the gap between advanced speech technologies and their effective deployment in web applications.***

Index Terms: Voice User Interfaces, Web Applications, Speech Recognition, Accessibility, HCI, Usability

I. INTRODUCTION

The swift progress of technology has drastically altered the way humans interact with computers, moving from command- line and graphical user interfaces (GUIs) to more natural and intuitive approaches such as voice-based systems. Voice User Interfaces (VUIs), which are driven by artificial intelligence and natural language processing (NLP), facilitate interaction through spoken commands, providing a hands-free, accessible, and efficient way to engage with digital systems [1].

Integrating voice control into web applications creates new opportunities for accessibility, usability, and productivity, as it enables users to complete tasks without relying on traditional input devices such as keyboards, mice, or touchscreens [2]. The global adoption of virtual assistants like Siri, Alexa, and Google Assistant has demonstrated the potential of voice in- teraction in daily activities. However, web applications, which serve as the primary platform for information and services, have been slow to adopt these features. By incorporating speech recognition capabilities, web applications can become more inclusive for users with visual impairments and motor disabilities, while also enhancing interaction efficiency for the general population [3].

Research in this field has evolved from voice navigation in structured web environments and speech-driven web browsing systems [4] to extensive surveys on web-based voice recog- nition applications. Meanwhile, commercial voice assistants such as Amazon Alexa, Google Assistant, and Apple Siri have popularized the use of Voice User Interfaces (VUIs), paving the way for innovative applications in accessibility, usability, and user experience evaluation [5].

Despite notable advancements in speech recognition tech- nologies, several challenges remain, including sensitivity to noise, accuracy issues, multilingual support, and privacy con- cerns. Additionally, while voice-enabled systems are available as standalone devices, their seamless integration into web applications is still lacking. Furthermore, although usability studies have been conducted, there is still an absence of context-dependent UX testing frameworks for VUIs [6].

This research, therefore, seeks to investigate the scope, methodologies, and challenges involved in developing voice- controlled web applications, assess existing frameworks, and suggest improvements to enhance user accessibility and expe- rience.

II. LITERATURE REVIEW

The idea of integrating voice recognition into computing systems has existed for decades, beginning with early speech recognition systems and gradually evolving into modern Auto- matic Speech Recognition (ASR) technologies. These systems enable natural communication between humans and comput- ers, with accuracies now exceeding 90

Several frameworks and standards have been proposed to embed speech capabilities in web environments. Microsoft's Speech Application Language Tags (SALT) introduced an XML-based markup language designed for multimodal dia- logue systems, enabling developers to integrate speech func- tionalities alongside graphical interfaces [7]. Although SALT influenced early developments, it was eventually succeeded by VoiceXML, an industry standard for interactive voice dialogues.

More recently, the Web Speech API and JavaScript libraries such as annyang.js have enabled developers to implement speech recognition directly into websites [8].

Research comparing commercial and open-source speech recognition platforms highlights the strengths of cloud-based APIs. Studies show that Google's Speech API and Microsoft's Speech API achieve superior performance compared to open- source alternatives such as CMU Sphinx, particularly due to their use of deep learning and neural network-based acoustic models [9]. These findings emphasize the reliability of commercial APIs in building practical voice-controlled web solutions.

In terms of application, voice automation has been demon- strated in multiple domains. For example, news portals inte- grated with Alan Studio and React allow users to navigate and access news content entirely through voice commands [3]. Similarly, prototypes using HTML5 speech recognition capabilities have shown that voice can replace traditional text input methods, such as filling forms or navigating websites [10].

Beyond technical feasibility, the usability and user ex- perience of VUIs are critical. A recent systematic review highlights that while VUIs provide unique benefits such as hands-free accessibility, they are often associated with us- ability issues like recognition errors and unnatural dialogue flows [12]. Effective design, therefore, requires balancing technical accuracy with user-centric considerations such as personalization, privacy, and error prevention.

A. Speech Recognition for Web Browsing

Dongre et al. proposed a speech recognition–based browsing system using Mel-Frequency Cepstral Coefficients (MFCC) for feature extraction and a Centroid-based Neural Network Adaptive Resonance Theory (CNN-ART) model for clas- sification [4]. Their system achieved 70–85% accuracy in recognizing words and sentences, demonstrating the feasibility of applying speech recognition models for browsing while also revealing limitations in noisy, speaker-independent settings.

B. Voice Interfaces in Accessibility and HCI

Recent studies emphasize the accessibility benefits of voice interfaces. Jha et al. analyzed how voice-based user interfaces (VUIs) enhance access for people with disabilities, particu- larly in scenarios where graphical user interfaces (GUIs) are limiting [9]. Similarly, Klein et al. explored how the context of use (task, environment, and user characteristics) influences user experience (UX) in VUIs [5]. They argue that context- sensitive evaluation frameworks are necessary to capture the nuances of real-world interaction.

C. Voice in Survey Methodologies

Beyond browsing and accessibility, Park et al. applied web- based voice recognition applications in survey research, posi- tioning them as alternatives to face-to-face interviews [12]. This approach draws on Audio Computer-Assisted Self- Interviewing (ACASI) methods, thereby enhancing privacy and efficiency. However, response rates were lower compared to traditional interviews, pointing to the need for improved engagement mechanisms.

D. User Experience (UX) Measurement for VUIs

Evaluating UX in VUIs has become a significant research focus. Klein introduced the concept of a UX tool selector that recommends suitable UX measurement methods based on context-dependent interaction scenarios [6]. This work highlights the absence of standardized UX evaluation frame- works for VUIs and emphasizes the importance of combining qualitative and quantitative approaches.

E. Research Gaps

Although voice technologies are increasingly integrated into devices and applications, their adoption in web applications remains limited. Current systems often focus on single do- mains or devices, lacking generalized frameworks for broader web usage. Additionally, there is insufficient research on evaluating long-term user experience, accessibility for diverse populations, and handling challenges like multilingual support and noisy environments.

III. METHODOLOGY

The methodology of this research is divided into four main phases: (1) Framework Selection, (2) Prototype Development, (3) Evaluation, and (4) Analysis of Results.

A. Framework Selection

To evaluate the feasibility of voice-controlled web applica- tions, existing speech recognition technologies were studied. Both

commercial APIs (Google Speech API and Microsoft Speech API) and open-source alternatives (CMU Sphinx) were considered. The Web Speech API was selected as the primary framework due to its native support in modern browsers and its seamless integration with JavaScript libraries such as annyang.js.

B. Prototype Development

A prototype web application named VoiceVista was devel- oped to demonstrate voice-based interaction within modern web environments.

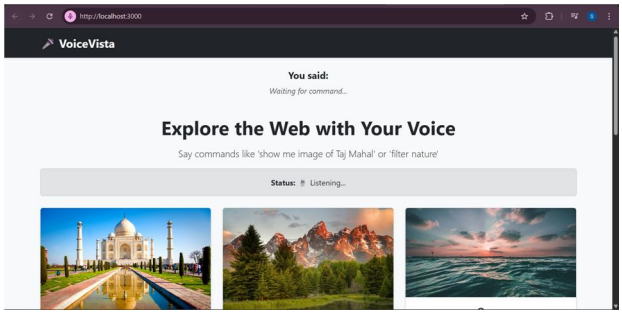


Fig. 1. Default listening state of the VoiceVista application.

The system leverages the Web Speech API to enable natural voice recognition and speech synthesis directly within the browser. It is integrated through the annyang.js library for speech-to-text processing and the SpeechSynthesis API for providing spoken feedback. The primary objective of the prototype was to explore hands-free, accessible, and intuitive interaction models by integrating voice commands into common web functionalities. The developed prototype incorporated the following core com- ponents:

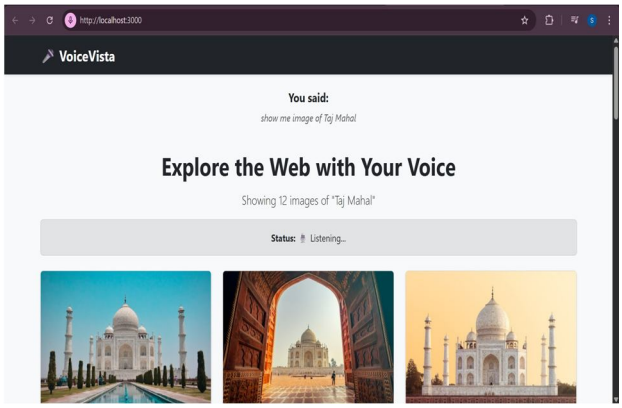


Fig. 2. Voice-enabled image search results for the command “show me image of Taj Mahal,”.

- Voice-based navigation: Users could access and switch between different sections of the application using natural voice commands such as “Scroll down,” “Scroll up,” and “Open Google dot com.”
- Form input through speech: VoiceVista supported dic- tation and speech-based entry into text fields, allowing users to fill out short forms or provide input through spoken commands instead of typing.
- Interactive feedback: The frontend provided real-time visual and audible responses using the SpeechSynthe- sisUtterance interface, allowing the application to “speak” confirmations such as “Showing images of Taj Mahal” or “Background changed to blue.”

The frontend architecture was implemented using HTML5, CSS3, JavaScript (ES6), and React.js, with Bootstrap used for responsive design. Voice functionalities were embedded directly via the Web Speech API, enabling smooth and low- latency performance in supported browsers.

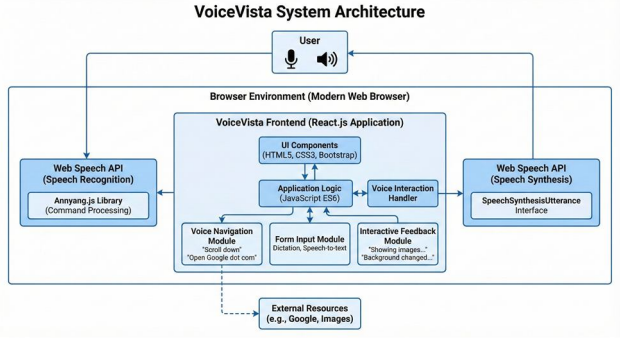


Fig. 3. System Architecture of VoiceVista.

C. Evaluation

The VoiceVista prototype was evaluated across three pri- mary dimensions to assess its effectiveness and practical viability for voice-driven interaction:

1) Accuracy: Speech recognition accuracy was measured using Word Error Rate (WER) across three environ- mental conditions—quiet, moderate noise, and high noise—representing typical user contexts.

2) Usability: Usability was assessed through a structured user study in which participants performed predefined tasks such as navigating between sections, filling a short form using voice, and retrieving multimedia content using spoken commands.

3) Accessibility: The system was tested with participants with visual or motor impairments to evaluate whether voice input could effectively replace traditional mouse or keyboard-based interactions for core web tasks.

D. Analysis of Results

Data from system testing and user studies were analyzed to examine the accuracy, usability, and accessibility performance of VoiceVista, as well as to identify design trade-offs associ- ated with browser-based voice interfaces.

The study compared the prototype's use of the Web Speech API with commercial cloud-based Automatic Speech Recog- nition (ASR) systems. While cloud APIs generally offer su- perior accuracy due to large-scale neural models and server- side computation, the Web Speech API provided real-time, low-latency performance suitable for in-browser interactions without external dependencies.

Errors were most frequent with proper nouns, acronyms, and uncommon terms, as well as during ambient noise interference. Despite these limitations, the system demonstrated stable recognition in quiet conditions.

The findings highlight that while the Web Speech API provides an effective and rapid development platform for browser-based voice interfaces, performance can degrade in noisy environments. However, its ease of integration and local execution make it highly suitable for prototyping and accessibility-focused applications like VoiceVista.

IV. RESULTS

A. Participants and Setup

A total of 30 participants (18 males and 12 females) with an age range of 19–55 years (mean ≈ 31) took part in the evaluation of the VoiceVista prototype. Among these, six par- ticipants self-identified as having visual or motor impairments, enabling the assessment of accessibility-focused use cases. All testing was conducted on desktop browsers, specifically Google Chrome and Microsoft Edge, both of which provide native support for the Web Speech API.

The evaluation tasks were designed to mirror real-world scenarios and followed the methodology established during the system design phase. Participants performed voice-based navigation, such as opening or switching sections and scrolling through the interface using commands like "Scroll down" or "Scroll up."

The study included voice-driven content retrieval, allowing users to fetch and display dynamic multimedia content—for instance, saying "Show me image of Taj Mahal" to retrieve images from the Unsplash API integrated within VoiceVista.

B. Accuracy (Word Error Rate — WER)

The Word Error Rate (WER) was calculated by comparing the transcriptions generated by the Web Speech API (accessed via Annyang.js) with the ground-truth speech transcripts provided by participants. The results demonstrated that the system performed well under quiet conditions, achieving a WER of 7.9% (SD = 3.1%). However, accuracy declined as background noise increased, with a WER of 17.6% (SD = 5.8%) in moderately noisy environments and 33.8% (SD = 8.9%) under high-noise conditions.

These results show a predictable degradation in accuracy as ambient noise levels rose. In quiet conditions, the recognition accuracy was sufficient for most interactive tasks, including navigation and information retrieval. In contrast, high noise levels noticeably affected recognition quality, underscoring the noise sensitivity of browser-based speech recognition systems like the Web Speech API.

C. Comparison: Web Speech API vs. Commercial ASR Systems

A comparative analysis was conducted between the Web Speech API—the engine underlying VoiceVista—and com- mercial Automatic Speech Recognition (ASR) systems offered by major cloud providers. The comparison revealed several key trade-offs.

The Web Speech API operates natively in web browsers and can be implemented directly using JavaScript, providing low- latency performance and simple integration without the need for external APIs or authentication processes. This makes it ideal for rapid prototyping, educational projects, and accessi- ble user interfaces such as VoiceVista. However, it is more sensitive to background noise and offers lower recognition accuracy compared to commercial ASR solutions.

By contrast, cloud-based ASR systems—such as those provided by Google, Microsoft, or Amazon—leverage pow- erful server-side deep learning models, resulting in higher accuracy and robustness across diverse acoustic environments. The trade-off, however, lies in increased latency, complex integration requirements, and potential privacy concerns due to speech data being transmitted and processed remotely.

Overall, the evaluation confirmed that while the Web Speech API may not match commercial systems in noisy condi- tions, it remains an excellent tool for browser-native voice interaction. Its simplicity, responsiveness, and full integration within client-side web technologies make it an ideal choice for prototyping accessible and interactive systems like Voice- Vista, where real-time feedback and user engagement are key priorities.

V. DESIGN GUIDELINES

From results and literature, here's a concise set of actionable guidelines for developers and designers of voice-enabled web applications:

1) Offer a clear activation model: Use a push-to-talk button or explicit activation phrase rather than continuous always-listening by default. This improves privacy and reduces false activations.
2) Provide concise, contextual feedback: Confirm recognized commands with short confirmations ("Opening X — say 'yes' to confirm"), rather than long spoken transcripts
3) Optimize for noisy environments: Use automatic gain control, Web Audio API prepro- cessing (noise suppression), and allow user selection of push-to-talk vs always listening.
4) Design predictable command grammars: For navigation, keep commands short and distinct (e.g., "Go to Home", "Open Articles") and avoid overloaded natural-language intents unless backed by robust NLU.

VI. CONCLUSION

This research demonstrates that voice-enabled web appli- cations are both feasible and beneficial — particularly for navigation and content retrieval tasks and for improving acces- sibility — but also that important challenges remain: ambient- noise robustness, reliable error correction, user trust / privacy, and consistent cross-browser behavior. The Web Speech API allows rapid prototyping and reasonable performance in quiet settings, but production-grade deployments should consider hybrid approaches (local pre-processing + optional cloud- based ASR), robust UX for error handling, and strict privacy controls.

VII. FUTURE WORK

Suggested directions to extend this research:

1) Larger-scale field study across diverse environments (outdoor, commute, public transport) and on mobile devices.
2) Compare multiple speech backends (browser Web Speech API vs Google Cloud Speech-to-Text vs Mi- crosoft Speech Services vs on-device models) in controlled and in-the-wild settings to quantify la- tency/accuracy/privacy trade-offs.

3) Adaptive models and personalization: incorporate on- device speaker adaptation, user-specific language mod- els, and incremental learning to improve recognition for named entities and accents.

4) Standardized VUI UX evaluation toolkit: develop and validate a context-sensitive UX testing framework for web VUIs (building on Klein (2021)'s ideas).

REFERENCES

[1] M. Deshmukh and R. Chalmeta, "User experience and usability of voice user interfaces: A systematic literature review," Research Group on Systems Integration and Re-Engineering (IRIS), Dept. of Languages and Computer Systems, Universitat Jaume I, Castello´n de la Plana, Spain.

[2] P. Verma, R. Singh, and A. K. Singh, "A framework to integrate speech- based interface for blind web users on the websites of public interest," Proc., details not provided.

[3] R. Kumar and A. Kumar, "Voice controlled news web application with speech recognition using Alan Studio," Dept. of Computer Science and Technology, Meerut Institute of Engineering and Technology, Meerut, India

[4] V. Dongre, S. Ghosh, and A. Katkar, "Speech recognition-based web browsing," EXTC Dept., Thakur College of Engineering and Technology (TCET), Kandivali (E), and Vidyavardhini's College of Engineering and Technology (VCET), Vasai (W), Maharashtra, India.

[5] A. M. Klein, J. Deutschla¨nder, K. Ko¨lln, M. Rauschenberger, and M. J. Escalona, "Exploring the context of use for voice user interfaces: Toward context-dependent user experience quality testing," Univ. of Seville, Spain; Univ. of Applied Sciences Emden/Leer, Germany

[6] A. M. Klein, "Toward a user experience tool selector for voice user interfaces," in *Proc. Conf. Human–Computer Interaction*, Apr. 2021, doi: 10.1145/3430263.3456728.

[7] V. Ke¨puska and G. Bohouta, "Comparing speech recognition systems (Microsoft API, Google API, and CMU Sphinx)," in *Proc. IEEE*, Florida Institute of Technology, USA.

[8] C. J. Varenhorst, *Making Speech Recognition Work on the Web*, M.Eng. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA.

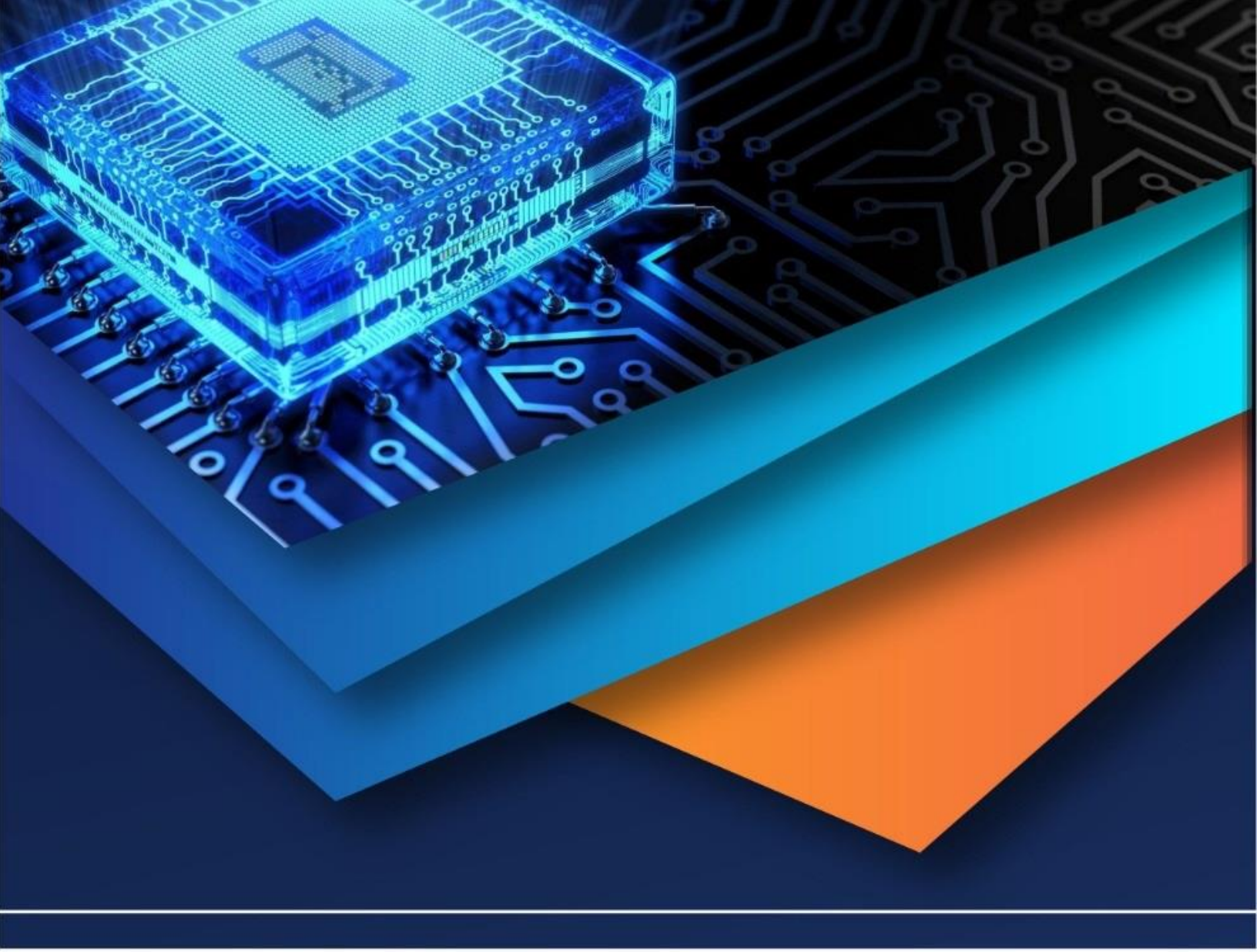
[9] R. Jha, M. A. M. Hassan, M. F. H. Fahim, and C. Rai, "Analyzing the effectiveness of voice-based user interfaces in enhancing accessibility in human–computer interaction," in *Proc. Int. Conf. CSNT*, Apr. 2024, doi: 10.1109/CSNT60213.2024.10545835.

[10] M. S. Kandhari, F. Zulkernine, and H. Isah, *A Voice Controlled E- Commerce Web Application*, Queen's University, Kingston, Canada.

[11] A. Al-Saif, "A systematic literature review to determine the web acces- sibility issues in Saudi Arabian university and government websites for disabled people," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 7, pp. 1–8, Jul. 2017

[12] Y. Park, T. Maeda, and D. Mochihashi, "Web-based voice recognition survey apps as the 'new interviewer': Development and practicality of a voice pseudo-face-to-face survey system," *J. Voice-Interaction Research*, vol. 4, no. 2, pp. 1–12, Nov. 2024.

[13] J. Zimmerman, "Case for a voice-internet: Voice before conversation," HCI Institute, Carnegie Mellon University, Pittsburgh, PA, USA.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY