IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 1, March 2025

AI Voice-Bot

Mr. Abhishek Mane[1], Mr. Vedant Barve[2] , Mr. Shubham Jadhav[3], Prof. Mohan Mali[4]
Lecturer, Department of Computer Technology Engineering[4]
Student, Department Computer Technology Engineering[1,2,3]
Bharati Vidyapeeth Institute of Technology, Navi Mumbai, India

Abstract: *In this paper, we present an AI-based desktop voice assistant built with Electron, that provides a natural conversation experience with integrated system-control capabilities on Windows OS. It listens to them, thanks to Wit, which processes your voice and text inputs. ai to transcribe the audio into reliable speech-to-text output and the Gemini AI API to respond in a coherent and context-appropriate manner. In addition to the conversational component, the system also boasts a new feature set enabling the user to control a variety of PC functions (like enabling/disabling WiFi, controlling the screen brightness & toggling the Bluetooth, etc.) via IPC handlers and native Windows commands. It is a modular architecture, owning separate pieces of code for authenticating users, processing the voice, making API calls and executing commands on the Machine. A secure authentication module to interact with a MySQL database to manage user credentials, while continuing voice recording pipelines with audio transformation routines ensures speech recognition high fidelity. Our empirical evaluations show that the combination of diverse APIs and system controls present a real-time and dynamic user experience, overcoming the hurdles of not synchronizing asynchronous interfaces. In the future, our focus will be on reducing the latency, improving the control of the system and the security protocols, which will make this more similar to desk assistants.*

Keywords: AI Voice Assistant, Electron desktop application, speech-to-text conversion, system control automation, Gemini AI API, Wit. Some of them are: ai Integration, Inter-Process Communication, User Authentication

I. INTRODUCTION

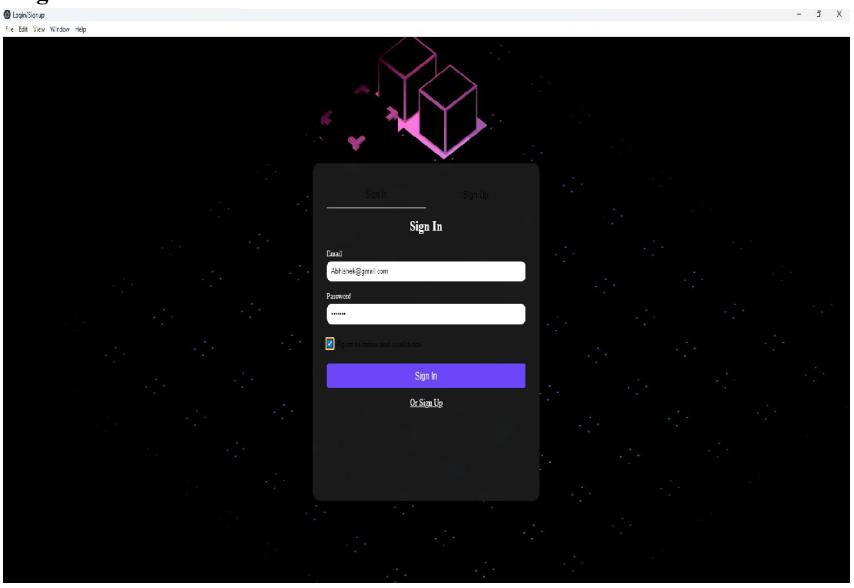
Artificial intelligence (Ai) and natural language processing (NPL) are enabling natural, humanlike interactions with voice assistants. voice interfaces are now widely used on smartphones, smart speakers, and now also on desktops to improve productivity and user experience. this paper presents a desktop voice assistant built with electron that takes natural language queries through advanced speech recognition and executes system-level commands on windows. it leverages cutting edge apis and tech. for voice recognition, it uses wit. ai to accurately transcribe speech. a real-time voice recording pipeline captures and processes audio data. the text is sent to gemini ai api, which generates contextually relevant and coherent responses. the system's architecture ensures a seamless conversational experience, managing asynchronous audio processing and api communication complexities. besides conversation, the assistant excels when managing desktop features. thanks to ipc in electron, it bridges the gap between high-level ui and low-level system commands. it executes native windows commands like toggling wifi, adjusting screen brightness, and managing bluetooth, delivering a seamless solution that combines intelligent conversation with practical system automation. this dual capability addresses a gap in existing desktop assistants, which either focus on communication or system control, but not both. user authentication is also needed and the app provides a secure login/signup that uses a mysql database to store user credentials which allows access only to authorized users for all functionality. this is important for apps that interact with system commands so that unauthorized access and abuse of control features is prevented. summing up in this paper, we have described the design, implementation, and experimental evaluation of an ai-based desktop voice assistant. through the integration of high-performance speech recognition, adaptive response generation, and system control into a single electron-based platform, the project offers a novel approach to enhance desktop interactivity. the following sections discuss the system architecture, underlying code, experimental results, and future improvement and scalability opportunities.

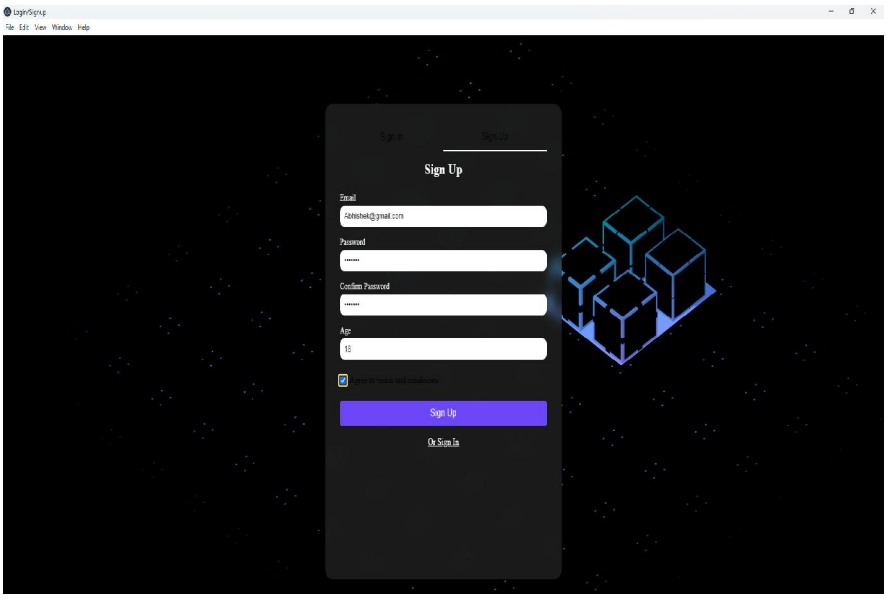
IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 1, March 2025

II. PAGE LAYOUT

A good page layout makes the content easier to read and more usable. There are two main pages.

A. Authentication Page





Design: Eye-catching layout with 3D background (via external Spline viewer) that grabs users attention at launch.

Components:

Toggle Buttons: Switch between "Sign In" and "Sign Up" forms.

Forms:

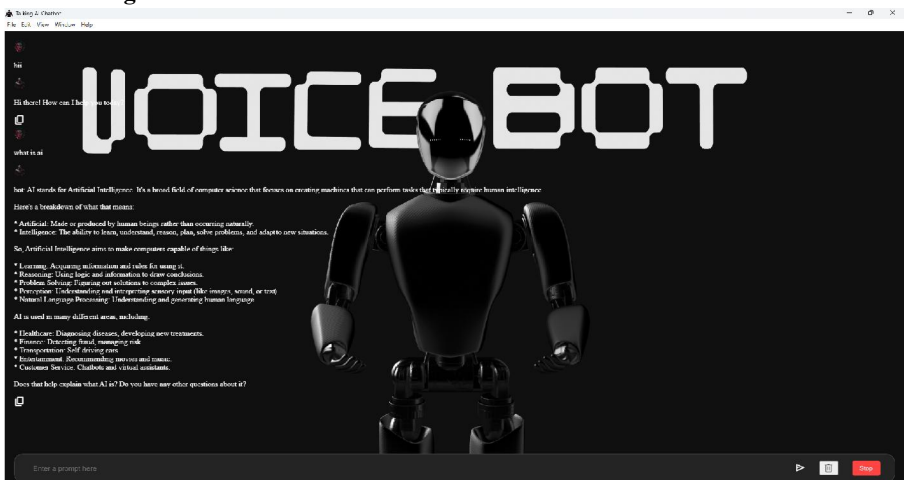
The sign in form contains email and password fields with feedback on input (red borders for errors).

Sign Up form collects email, password, password confirmation and age information.

The implementation is described in index. html and controlled by script1. js which toggles the form and validates the input.,).

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 1, March 2025

B. Chatbot Interaction Page



Design Once a user has authenticated, they'll be taken to a dedicated chatbot interface.

Components:

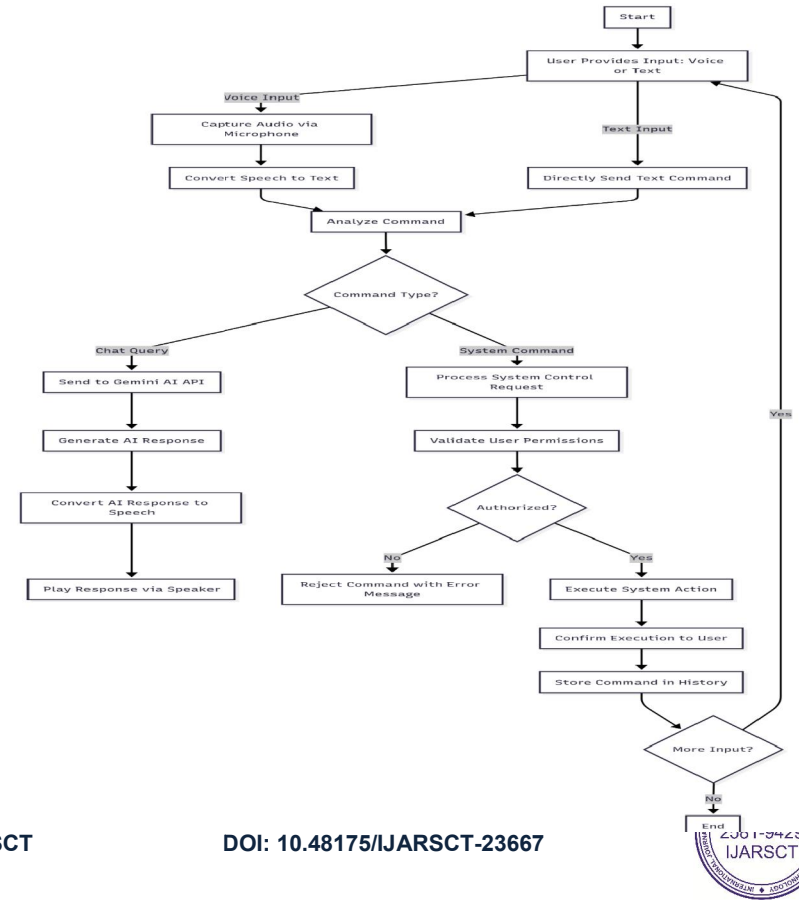
A dynamic header and Spline 3D background gives this page a modern look.

Chat Area: Shows conversation history with separate chat bubbles for user and assistant messages.

Typing area: A text box, a send button, and buttons for deleting chats, stopping voice processing, and so on.

Layout is defined in AiChatbot. html, and Scriptsai. js handles dynamic behavior like continuous voice recording, live chat updates, and stops.

III. WORKING PROCESS



IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 1, March 2025

User Input

- The process begins when a user provides input (either by speaking or typing).

Convert Speech to Text (if needed)

- If the user spoke their request, the system first turns it into text.

Send Request to ChatGPT

- The text request is then sent to ChatGPT (via a command).

Get and Process ChatGPT's Response

- ChatGPT replies with its answer.
- The system takes that answer and prepares it for output (e.g., checks formatting, etc.).

Check for Commands / Authentication

- If the response or the user's request includes a special command (like an instruction to run a specific action), the system decides whether it's allowed:
- If it needs extra permission, it goes through an "Authorizer."
- If something looks suspicious (like a command injection error), the system may log it or stop.

Optional: Re-inject Command

- If the user (or the system) wants to feed the AI's response back into ChatGPT for further processing, it injects the new command and processes that as well.

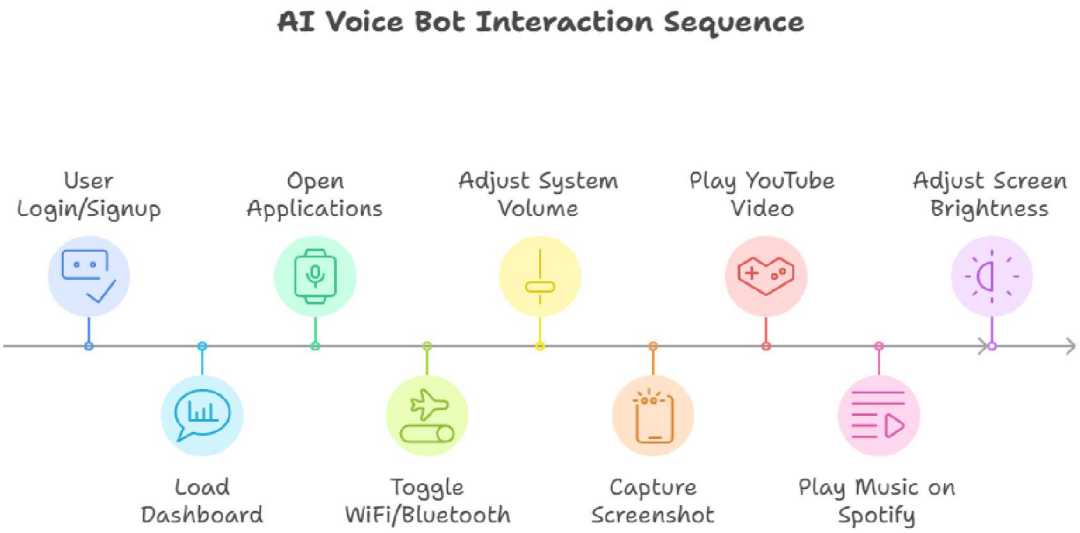
Output the Final Response

- Once everything is cleared and processed, the system either:
- Displays the text result, and/or
- Plays it aloud (using text-to-speech).

End / Log Activity

- The system stores the history of commands and responses.
- The flow ends.

IV. FEATURES

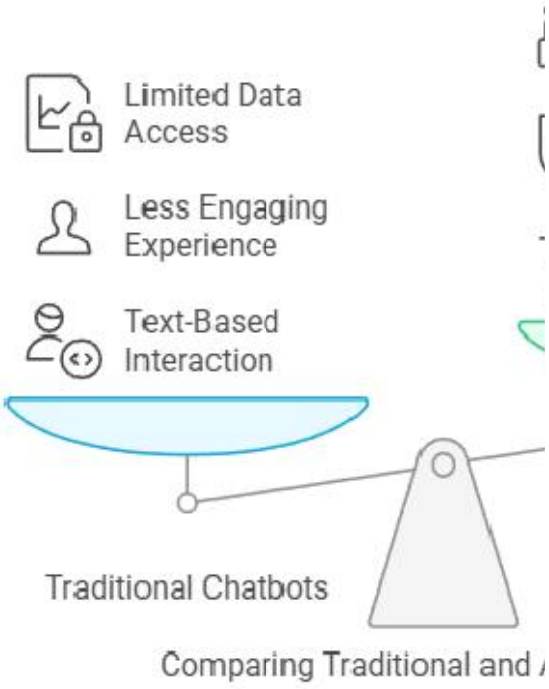


- **Login/Signup Screen**: The screen offers a secure environment to sign up to for first-time users and log in with authorized credentials for repeated users.
- **Main Dashboard**: The personalized dashboard is unveiled at login time, and it has a text and voice command chat window.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 1, March 2025

- **The AI Voice Bot** enables one to open apps such as VS Code or Notepad and edit files or debug code using voice commands.
- **Toggle WiFi/Bluetooth**: Looks for voice commands to turn WiFi or Bluetooth on/off, runs system commands, and monitors changes.
- **Adjust Volume**: System volume can be adjusted by voice commands associated with the controls, with confirmation by feedback.
- **Take Screenshots**: The bot captures a screenshot with the system function and gives both visual and audio feedback.
- **Play YouTube Videos**: It opens YouTube, constructs search queries if needed, and plays the first voice command result.
- **Play Music on Spotify**: The bot opens Spotify, identifies play or search commands for a song, and plays the song.
- **Adjust Screen Brightness**: It manages brightness via commands such as "set brightness high" or "set brightness low."
- **File Operations**: It outputs to a file, edits it in an editor such as VS Code or Notepad, and gives feedback.
- Open Social Media: It opens given social media pages from the command in the default browser with instant feedback.
- **Stop/Restart or Shutdown:** The bot shuts down its operations gracefully or initiates the restart/shutdown process when commanded.

V. PROBLEM SOLVING



IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 1, March 2025

1. Need for Hands-Free Interaction:

- Many applications today require typing or clicking, which can be cumbersome in situations where hands-free operation is needed. For example, users may need assistance while driving, cooking, or multitasking.
- Voice bots provide a solution by allowing users to interact through speech, making it more accessible and convenient.

2. Enhanced User Experience:

- Traditional chatbots often rely solely on text-based input and output, which can be less engaging and slow.
- An AI voice bot can make the interaction more dynamic and user-friendly by enabling speech recognition and voice responses, offering a more natural way of communicating with machines.

3. Efficient Data Management:

- Many users find it frustrating when they can't access past conversations with AI assistants or bots.
- By storing chat data locally on the user's PC, this voice bot allows users to retrieve and review previous interactions, improving user satisfaction and allowing continuity in conversations

Benefits and Usefulness:

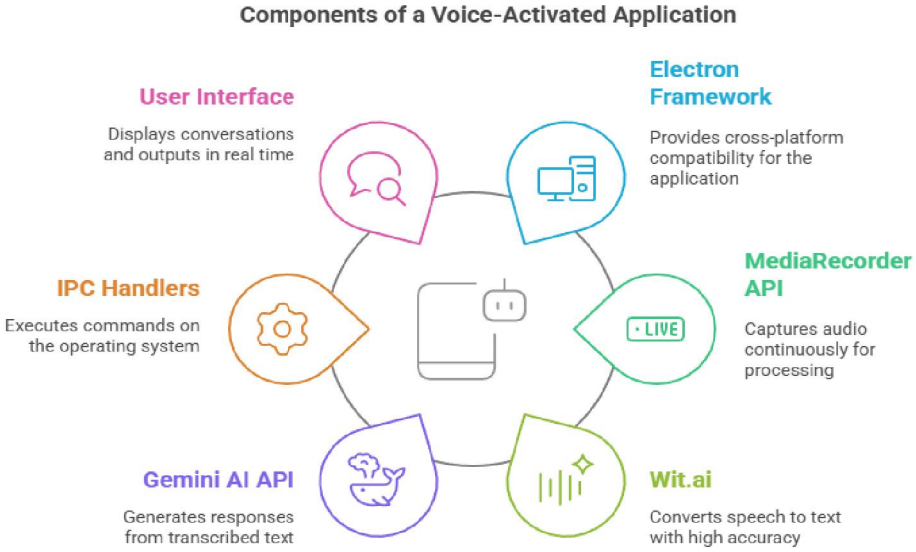
1. Increased Accessibility and Convenience:

- Voice bots make it easier for users to interact with technology, especially in environments where typing or using a touchscreen is impractical.
- They cater to a wide range of users, including those who may have physical disabilities or impairments that limit traditional interaction methods, enhancing inclusivity.

2. Real-Time Intelligent Responses:

- By connecting the bot to an AI system through an API, users can receive real-time answers to their queries. This makes the bot useful for various domains such as customer support, education, and personal assistance.
- The bot can offer immediate and relevant information, reducing wait times and improving user satisfaction

VI. METHODOLOGY



IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 1, March 2025

System Architecture:

- The application is built using Electron for a cross-platform desktop environment, featuring modular components for user authentication, voice processing, API communication, and system control.

Voice Data Acquisition:

- Audio is captured continuously using the Media Recorder API. The recorded data is converted into raw PCM format suitable for processing.

Speech Recognition & Processing:

- The raw audio is sent to Wit.ai for high-accuracy speech-to-text conversion. The transcribed text is then forwarded to the Gemini AI API to generate relevant responses.

System Control:

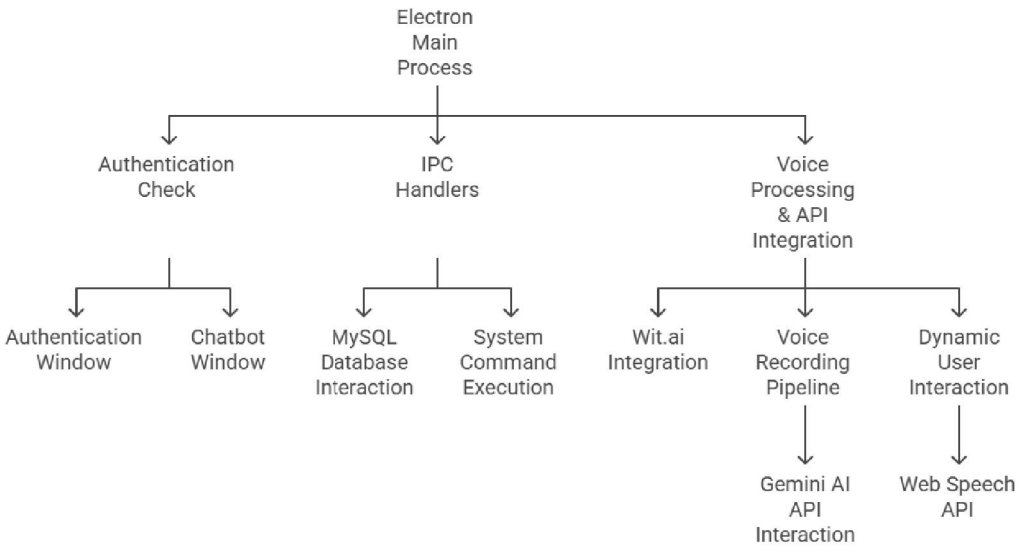
- Extracted commands are interpreted and executed on the Windows operating system using IPC handlers and native system commands (e.g., toggling WiFi, adjusting brightness).

User Interface:

- A dynamic chat interface built with HTML, CSS, and JavaScript displays the conversation in real time and uses the Web Speech API for text-to-speech output.
- **Evaluation:**
  Performance is assessed by measuring latency, transcription accuracy, and user interaction, with adjustments made based on feedback.

VII. SYSTEM ARCHITECTURE & CODE ANALYSIS

Application Framework Flowchart



A. Core Application Framework (Electron Main Process)

Overview:

The application logic is in the main. js file, where the Electron framework initializes the application, creates application windows, and handles IPC (Inter-Process Communication) so you can communicate easily between the renderer process and the main process and also check for the existence of an authentication cookie via Electron's session API.

B. Voice Processing & API Integration

Wit.ai Integration:

The preload script (preload. js) initializes a Wit. ai client with the access token required to process voice data. The exposed function processVoice converts audio data into a text string using the Wit. ai speech recognition services.).

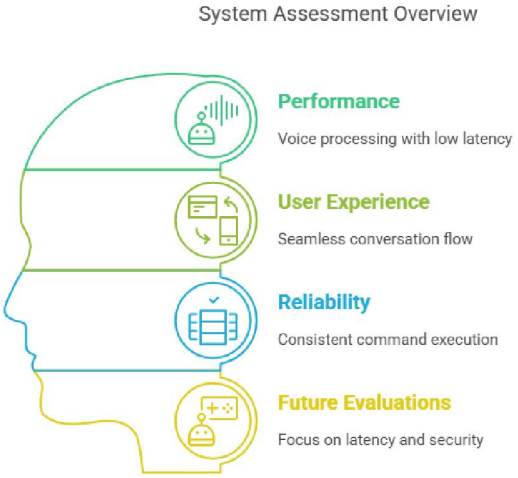
IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 1, March 2025

C. System Control Capabilities

IPC for System Commands:

IPC handlers are defined in main. js to run system commands like turning on WiFi or changing brightness levels. These handlers use Node. js modules like child_process and exec to reliably run Windows native commands, with multiple fallback methods if needed.

VIII. EXPERIMENTAL EVALUATION



Performance:

Takes voice continuously and processes it in a pipeline with very low latency for response times close to real-time.

User Experience:

The dynamic updates in the UI, such as live chat and speech synthesis, make for a seamless conversation flow.

Reliability:

Native Windows commands executed like a champ, and fallbacks were in place, keeping the behavior consistent across system configurations.

IX. CONCLUSION

In this paper, we present the construction of an AI-powered desktop voice assistant that incorporates speech recognition capabilities, conversational AI facilities, as well as realistic dynamic system control attributes, all held together in an Electron-based workspace. The system effectively combines heterogeneous APIs and IPC through a modular design approach to provide a seamless user experience. Asynchrony, latency, and multi-sites remain big challenges, but empirical evaluation shows the promise of the system. Additional efforts will improve performance, expand system control capabilities, and secure protocols.

X. ACKNOWLEDGMENT

The authors sincerely appreciate the support and guidance from colleagues and mentors who offered valuable insights and feedback throughout the project's development. A special thank you goes to the developers of the Electron framework, as well as the teams at Wit.ai and the Gemini AI API, for providing essential tools in free.

REFERENCES

[1]. Wit.ai Meta Platforms, Inc. (n.d.). Wit.ai Documentation. Retrieved from https://wit.ai/docs (For intent recognition, speech-to-text, and NLP workflows.)

[2]. Gemini AI Google AI. (2023). Introducing Gemini: Google's next-generation AI model. Retrieved from https://blog.google/technology/ai/(Cite this for Gemini's generative capabilities and API integration.)

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.67
Volume 5, Issue 1, March 2025

[3]. Devlin, J., et al. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL. (Background on transformer-based NLP models, relevant for Gemini's architecture.)

[4]. Microsoft. (2023). Inter-process Communication (IPC) in Windows. Retrieved from https://learn.microsoft.com/en-us/windows/win32/ipc (For IPC mechanisms like pipes, sockets, or RPC used for OS control.)

[5]. pywin32/pywinauto Python Software Foundation. (n.d.). Python for Windows Extensions. Retrieved from https://pypi.org/project/pywin32/ (If you used Python libraries to interact with Windows APIs.)

[6]. Zhang, Y., et al. (2020). A Survey of AI-Driven System Automation. IEEE Transactions on Human-Machine Systems. (For academic context on AI controlling OS-level tasks.)

[7]. Text-to-Speech (TTS) Ribeiro, F., et al. (2021). Neural Text-to-Speech: A Systematic Review. ACM Computing Surveys. (Covers TTS frameworks like pyttsx3 or gTTS used in your project.)

[8]. Speech Recognition Amodei, D., et al. (2016). Deep Speech 2: End-to-End Speech Recognition in English and Mandarin. Proceedings of ICML. (Foundational work on speech-to-text systems, relevant to Wit.ai.)

[9]. Shneiderman, B. (2022). Designing the User Interface: Strategies for Effective Human-Computer Interaction. Pearson. (For UI/UX principles in voice/text-based bots.)

[10]. Luger, E., & Sellen, A. (2016). "Like Having a Really Bad PA": The Gulf between User Expectation and Experience of Conversational Agents. CHI Conference on Human Factors in Computing Systems. (Highlights challenges in designing conversational AI.)

[11]. Vaswani, A., et al. (2017). Attention Is All You Need. Advances in Neural Information Processing Systems (NeurIPS). (Seminal paper on transformers, the backbone of Gemini and similar models.)

[12]. OpenAI. (2023). GPT-4 Technical Report. Retrieved from https://arxiv.org/abs/2303.08774 (Compare Gemini's architecture with other LLMs.)