SAYLINK: A Voice-Controlled Social Media Web Application Using Speech Recognition, NLU and TTS for Accessible Interaction

Prof. Ranjana Singh[1], Vikas Gupta[2], Sahil Ghorpade[3], Sanket Dhage[4], Satyam Jha[5]
[1]*Project Guide, Watumull Institute of Engineering and Technology, Mumbai, India*
[2,3,4,5]*Computer Engineering, Watumull Institute of Engineering and Technology, Mumbai, India*

Abstract - Voice-driven interaction is emerging as a significant advancement in human computer communication, enabling hands-free operation and improving accessibility across digital platforms [1], [2]. Traditional social media interfaces rely on touch and typing, which may limit usability for multitasking users and individuals with motor impairments. This paper presents SAYLINK, a voice-controlled social media web application that integrates Speech-to-Text (STT), Natural Language Understanding (NLU), and Text-to-Speech (TTS) to execute actions such as liking posts, commenting, following users, searching content, and navigating between pages using spoken commands. The system architecture combines a React frontend for user interaction, client-side speech processing and command interpretation with a Node.js backend for application logic, and MongoDB for persistent data storage. The approach demonstrates a practical implementation of voice-driven interaction within a social media environment, aligning with voice-driven web interaction and command-based automation systems [3], [5]. Experimental evaluation demonstrates an average STT accuracy of 92.3% in quiet settings and 87.0% under moderate background noise, with an average response latency of 1.45 seconds, indicating efficient and reliable hands-free performance. SAYLINK highlights the potential of voice-based social platforms to enhance accessibility, convenience, and inclusivity while offering opportunities for future expansion to multilingual and adaptive NLU models [6].

Keywords - Voice Recognition, Web Speech API, React.js, Node.js, Express.js, MongoDB, Socket.IO, RESTful APIs, Event-Driven Architecture, Hands-Free Interaction, Social Media Interface, Accessibility.

I. INTRODUCTION

In recent years, social media platforms have evolved into essential tools for communication and content sharing, enabling users to connect, share content, and interact with online communities. Despite their widespread use, most interactions on these platforms still depend on manual actions such as typing, tapping, scrolling, and navigating through menus. These traditional interaction methods can create accessibility barriers for users with visual or motor impairments and may also reduce convenience for individuals who want to interact while multitasking. Voice-based interaction offers a more natural alternative that can simplify user engagement and improve accessibility across web applications [1], [2].

Although speech-enabled interfaces have seen adoption in areas such as search engines, virtual assistants, and assistive technologies, their use within dynamic social media feeds remains limited. Social platforms involve continuously updating content, multiple interactive elements, and context-dependent actions, making seamless voice control more complex to implement. Providing continuous voice interaction in such environments requires accurate command recognition, efficient event handling, and responsive interface updates [3], [4].

To address this gap, this work introduces SAYLINK, a voice-first social media web application that allows users to interact with social content using conversational speech. The system integrates real-time Speech-to-Text (STT) to capture spoken commands, Natural Language Understanding (NLU) to interpret user intent, and Text-to-Speech (TTS) to provide audible confirmation. SAYLINK follows a modular architecture consisting of a React-based frontend for user interaction, client-side speech processing and command interpretation with a Node.js backend connected to MongoDB for persistent data storage.

This modular structure enables independent enhancement of speech components, such as using browser-based speech recognition to ensure low-latency and privacy-preserving interaction [5], [6].

The motivation for developing the system is based on three primary objectives. First, the platform aims to improve accessibility by providing an alternative interaction method for users who may find conventional input devices difficult to use. Second, it enhances hands-free productivity by allowing users to perform common social media actions through voice commands while multitasking. Third, the system addresses an existing research gap by supporting continuous, multi-command voice interaction within a dynamic social media environment, extending beyond single-command use cases such as voice search or simple dictation.

This work makes several important contributions. It presents the design and implementation of a full-stack voice-controlled social media platform and introduces a rule-based intent recognition approach that combines rule-based command parsing with an optional machine learning fallback to improve robustness across different speaking styles. Additionally, the study includes a performance evaluation focusing on recognition accuracy, response latency, noise tolerance, and overall user experience. The results highlight both the usability benefits and the practical considerations involved in deploying voice-driven interaction in real-time social media systems.

II. RELATED WORK

Voice-enabled assistants such as Siri, Alexa, and Google Assistant have significantly improved hands-free interaction and natural language communication. However, their use within dynamic social media interfaces remains limited, particularly in scenarios requiring continuous interpretation of multiple sequential actions. Previous research on voice user interfaces (VUIs) has emphasized improving accessibility and simplifying interaction across web applications [1], [2]. Similarly, studies in Speech-to-Text (STT) technologies have focused on enhancing recognition accuracy and reducing processing latency under varying acoustic conditions [3].

Research in intent recognition and natural language understanding has further enabled spoken commands to be translated into executable system operations [4]. Despite these advancements, many web-based implementations using the Web Speech API are restricted to single-step tasks such as voice search or text dictation. These systems typically lack a complete interaction pipeline that includes intent recognition, backend execution, and real-time interface updates. The approach presented in this work addresses these limitations by enabling continuous, multi-intent voice control tailored specifically for social media interaction workflows.

III. SYSTEM OVERVIEW AND ARCHITECTURE

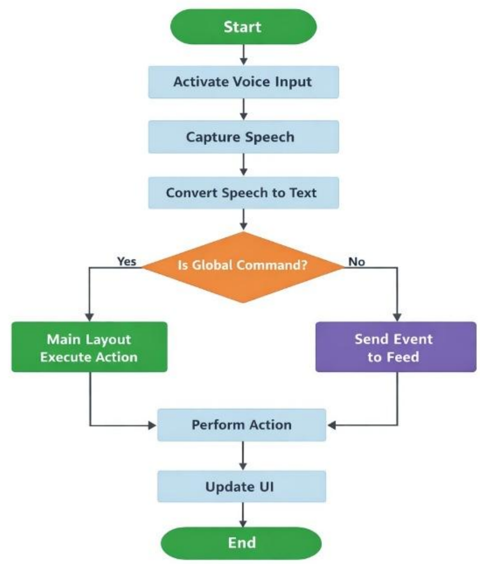

Fig. 1: Flowchart of SAYLINK

SAYLINK is designed using a modular architecture that integrates voice recognition with a social media interface to enable hands-free interaction. The system separates the frontend interface, voice processing module, backend services, and database layer to ensure scalability and maintainability. The workflow begins with voice input from the user, which is captured and converted into text using browser-based speech recognition. The processed command is then analyzed to determine whether it is a global navigation command or a context-specific action. Based on this classification, the system routes the command to the appropriate component, performs the required operation, and updates the user interface accordingly, as illustrated in Fig. 1.

The frontend of SAYLINK is developed using React.js, which provides a component-based structure for rendering feed posts, profiles, comments, and navigation controls. The Web Speech API is integrated into the client side to capture and process spoken commands in real time. A custom event-based communication mechanism is implemented to handle global voice commands across multiple components. Commands such as scrolling, liking posts, opening profiles, and adding comments are mapped to UI actions, allowing users to navigate the application without manual interaction. Styling is implemented using CSS and Bootstrap icons to maintain a responsive and consistent design.

The backend is implemented using Node.js and Express.js, which manage application logic, authentication, and RESTful API communication between the frontend and database. MongoDB is used as the primary data storage system to store user profiles, posts, comments, followers, and notifications. The flexible schema design of MongoDB supports dynamic social media content and enables efficient retrieval of data. Additionally, Socket.IO is integrated to provide real-time updates such as notifications, follows, and interaction events, ensuring that changes are immediately reflected in the user interface.

The architecture follows key design principles including modularity, low-latency interaction, and scalability. Frequently used commands are processed on the client side to reduce network overhead, while backend services ensure secure and consistent data handling. The system also supports graceful degradation, allowing traditional UI interaction if voice recognition is unavailable. This design provides a flexible foundation for future enhancements such as multilingual voice commands, improved intent detection, and accessibility-focused navigation.

IV. SPEECH RECOGNITION PIPELINE

The speech recognition pipeline in SAYLINK enables users to interact with the social media interface using voice commands. The pipeline begins when the user activates the microphone, after which the system captures spoken input through the browser. The captured audio is processed using the Web Speech API, which converts speech into textual commands in real time. This text output is then forwarded to the command processing module, where it is prepared for interpretation. The overall flow ensures minimal latency and supports continuous hands-free interaction within the application.

Once speech is converted to text, the command is normalized to improve matching accuracy. The normalization process includes converting the text to lowercase, removing extra spaces, and trimming unnecessary characters. This standardized command is then compared against a predefined set of supported voice actions such as “scroll down,” “like,” “comment,” and “open profile.” By using rule-based matching, the system quickly determines the user’s intent and maps it to the appropriate application behavior.

After intent detection, the system classifies commands into global commands and context-specific commands. Global commands, such as navigation and scrolling actions, are handled by the main layout controller to ensure consistent behavior across the entire application. In contrast, feed-specific commands, such as liking a post or adding a comment, are dispatched using a custom event-based communication mechanism. These events are broadcast across components, allowing the feed module to listen for relevant commands and execute the corresponding action.

The feed component receives the dispatched event and performs the requested operation, such as liking a post, opening a profile, adding a comment, or sharing content. Once the action is completed, the React state is updated, triggering a re-render of the interface. This ensures that user interactions are reflected immediately without requiring manual refresh. The event-driven approach enables seamless communication between independent UI components.

Finally, the system provides feedback to confirm successful execution of the command. Visual indicators such as toast notifications, status messages, or UI highlights inform the user that the action has been performed. This feedback mechanism improves usability and helps users maintain awareness of system responses during hands-free interaction. The complete speech recognition pipeline ensures responsive, accurate, and intuitive voice-based control within the SAYLINK platform.

V. NATURAL LANGUAGE UNDERSTANDING

The Natural Language Understanding (NLU) module in SAYLINK is designed to interpret transcribed speech and identify the user's intended action. To achieve both efficiency and reliability, the system adopts a hybrid approach that combines rule-based parsing with an optional machine learning fallback. This ensures that commonly used commands are processed quickly, while still maintaining the ability to handle variations in speech patterns or ambiguous inputs. The rule-based component handles most interactions with minimal delay, making it suitable for real-time web environments.

The rule-based parser relies on predefined keyword patterns and regular expressions to recognize actions such as liking a post, commenting, scrolling, navigating between pages, or searching content. These mappings are lightweight and optimized for speed, allowing immediate execution of frequent commands. In cases where the input does not clearly match predefined patterns, a machine

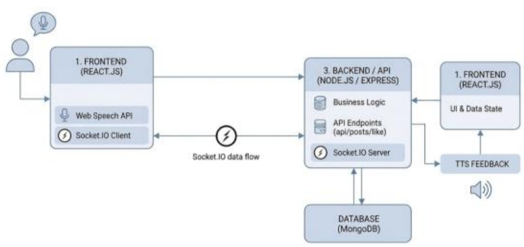

Fig. 2: System Architecture of SAYLINK

learning classifier is triggered as a fallback. This model, trained on sample voice commands, helps interpret less structured or conversational inputs, thereby improving system robustness across different speaking styles and accents.

SAYLINK defines a structured set of intents that correspond to key social media interactions. These include actions such as liking a post, adding comments, scrolling through the feed, following or unfollowing users, searching for content, navigating across pages, and opening specific posts. Each intent is directly mapped to a system function, enabling seamless execution of voice-driven commands. This predefined intent inventory ensures that users can perform essential social media operations efficiently using speech alone.

In addition to intent recognition, the system performs entity extraction to identify specific targets associated with each command. Instead of relying on complex natural language processing pipelines, SAYLINK uses lightweight heuristics to maintain responsiveness. For example, numeric inputs are mapped to post indices, usernames are detected based on common patterns or prefixes, and spoken phrases are interpreted as comment content when appropriately structured. This approach allows accurate extraction of relevant details while keeping processing overhead low.

Overall, the NLU module balances speed and flexibility by combining deterministic parsing with adaptive learning. This design enables SAYLINK to deliver accurate intent recognition and natural interaction while ensuring that performance remains suitable for real-time social media usage.

VI. ACTION EXECUTION LAYER

Once the user's spoken command has been converted into text and the corresponding intent and entities have been identified, the system generates a structured intent object that is passed to the action execution layer. This intent object is represented as JSON and contains the action type, the target entity (such as post ID or username), and any additional parameters such as comment text or scroll direction.

The frontend then maps this intent object to the appropriate backend API request. The Node.js backend receives this request, validates the operation, updates the data stored in MongoDB, and returns a structured response indicating success or failure. After the backend response is received, the React frontend updates the user interface to reflect the completed action—for instance, highlighting a liked post or appending a new comment in the feed.

To maintain a fully hands-free interaction experience, SAYLINK also uses Text-to-Speech (TTS) to provide audible confirmation feedback such as "Post liked," "Comment added," or "Scrolling down." This creates a closed interaction loop where the user speaks a command, the system executes it, and the user receives both visual and spoken acknowledgment. The combined use of UI updates and TTS ensures high interaction clarity and reinforces user confidence in voice-based control.

VII. USER INTERFACE AND UX DESIGN

The user interface in SAYLINK is designed to support natural, hands-free interaction while maintaining clarity and user control. The layout is based on a familiar social feed structure, ensuring that users transitioning from conventional platforms can interact without additional learning effort. Voice interaction elements are integrated in a way that makes the system state and feedback clearly visible, reducing confusion and helping the user remain oriented during continuous voice operation.

A key aspect of the interface is the listening state indicator, which informs the user whether the system is idle, actively listening, or processing a command. This prevents uncertainty about when to speak and aligns with best practices for voice-driven interaction. When an action is executed, the UI provides immediate visual confirmation, such as animating the like icon or appending a new comment to the post. Alongside visual feedback, short Text-to-Speech confirmations reinforce successful execution; these confirmations can be muted if the user prefers a silent interface.

The system also includes error-recovery prompts to ensure smooth conversational interaction. If a command is unclear or not recognized, the interface requests clarification or suggests re-phrasing, rather than failing silently. Despite the emphasis on voice input, traditional manual controls remain fully accessible at all times. This ensures that users can switch seamlessly to touch or mouse interaction when voice recognition is inconvenient or when network or environmental conditions reduce STT accuracy.

A. Keyboard Shortcuts

For users who prefer faster interaction without relying entirely on speech, SAYLINK includes optional keyboard shortcuts. In particular, the shortcut Ctrl + L toggles the listening mode, allowing experienced users to activate or pause voice recognition quickly without manually selecting the microphone icon.

VIII. IMPLEMENTATION DETAILS

The prototype implementation of SAYLINK is developed using a full-stack web architecture, where each layer is responsible for a distinct set of operations. The system is structured to allow independent improvement of the speech, intent, and interaction modules without modifying the rest of the platform.

The frontend is built using React.js, which provides a dynamic component-based UI for rendering the social media feed and user interactions. Speech recognition is handled using the Web Speech API, while audible confirmations are generated through the browser's SpeechSynthesis interface. Communication with backend services takes place over HTTPS using Axios, and the interface styling adopts claymorphism, which provides soft shadows, rounded shapes, and a visually clear hierarchy suitable for accessibility.

The middleware layer is responsible for handling command interpretation and coordinating voice-based interactions within the system. In the default configuration, transcripts generated by the browser-based speech recognition module are directly processed by the application's command handler, where the text is parsed to determine the user's intent and associated parameters. This approach reduces system complexity and avoids the need for an additional middleware service. Since speech recognition is performed on the client side using the Web Speech API, the system maintains low latency while ensuring that voice commands are interpreted consistently. This design prioritizes simplicity, responsiveness, and efficient execution of voice-driven actions without relying on a separate Flask-based processing layer.

The backend is developed using Node.js and Express, providing RESTful endpoints for actions such as liking posts, adding comments, following users, or retrieving content feeds. JSON Web Tokens (JWT) are used for authentication to ensure stateless session management and secure API access.

The database layer uses MongoDB, with structured collections for users, posts, comments, likes, and logs. Logs are stored to support accuracy measurement, intent error analysis, and future refinement of the NLU model.

This modular architecture allows SAYLINK to operate efficiently under normal conditions, while still being adaptable for experimentation with improved STT models, multilingual support, or additional intent types in future development.

IX. ALGORITHMIC DESIGN AND PSEUDOCODE

The command processing workflow in SAYLINK follows a structured pipeline that transforms user input (spoken or typed) into a successful UI action with confirmation feedback. The algorithm ensures that both speech-based and text-based interactions are handled uniformly.

The system begins by determining the format of the incoming input. If the user provides speech, the audio signal is passed through the selected Speech-to-Text engine. If the input is already text (e.g., through manual typing or accessibility tools), the transcription step is skipped. The resulting text is then normalized to handle lowercasing, filler word removal, punctuation smoothing, and common phrase correction.

Next, the transcript is passed to the intent parser, which produces an identified intent, the extracted entities (such as post index or comment text), and a confidence score that indicates how certain the parser is about the interpretation. When the confidence score falls below a predefined threshold, the system prompts the user for clarification to avoid unintended actions. This prevents accidental likes, follows, or comments, which are especially important when commands are performed hands-free.

Once a valid intent and entity set is confirmed, the system maps the command to the appropriate backend API call. The backend processes the request and returns a result, which is then used to update the interface in real time. Finally, the Text-to-Speech module provides a brief spoken acknowledgment to reinforce interaction awareness.

High-Level Algorithm Outline:

1) Input Handling:
   Determine whether the input is audio or text. If audio, perform speech recognition to obtain the transcript.
2) Normalization:
   Clean and standardize the transcript to reduce ambiguity.
3) Intent Parsing:
   Identify the user's intended action and any associated entities.
4) Confidence Evaluation:
   If the intent confidence is below threshold, request clarification; otherwise continue.
5) Action Execution:
   Map the intent to a backend API request and apply UI updates based on the response.
6) Feedback:
   Provide spoken and visual confirmation of the completed action.

This algorithm ensures that SAYLINK maintains both accuracy and interaction clarity, allowing the system to function effectively in real-time and hands-free usage scenarios.

X. EXPERIMENTAL SETUP

A. Dataset and Testbed

A dataset of 500 spoken commands was prepared, covering common interactions such as like, comment, scroll, search, follow, and navigation. To evaluate robustness, the same commands were also tested under background noise conditions at 10–20 dB SNR. Additionally, live spoken inputs were collected from 12 participants to reflect natural variation in speech patterns.

B. Evaluation Metrics

The system was assessed using the following metrics:

1) STT Accuracy: Complement of word error rate.
2) Intent Accuracy: Correct identification of command meaning.
3) Entity Extraction Accuracy: Correct extraction of post index, username, or comment text.
4) End-to-End Latency: Time from speech input to UI confirmation.
5) User Satisfaction: Measured using a 5-point Likert scale.

C. Hardware/Software Environment

Tests were conducted on a Windows 10 client using Chrome (v115+), with both built-in and external USB microphones. The server ran on a local VM (4 vCPU, 8 GB RAM). The backend services were hosted using Node.js and Express. Network conditions included localhost (0 ms) and a simulated 50 ms RTT to approximate typical Wi-Fi performance.

XI. RESULTS

Table I summarizes the Speech-to-Text (STT) accuracy, intent classification accuracy, and end-to-end latency across major interaction categories. The system achieves an average STT accuracy of 92.3% and intent accuracy of 95.0%, with an average response latency of 1.45 seconds, indicating that commands are processed reliably and within interactive time thresholds.

Table I — STT and Intent Performance (50 commands per category)

Category	STT Acc. (%)	Intent Acc. (%)	Latency (s)
Like/Unlike	94.3	97.1	1.18
Comment	90.5	92.0	1.62
Follow/Unfollow	91.8	93.5	1.38
Scroll / Navigation	95.7	98.0	1.08
Search	92.1	94.0	1.57
Average	92.3	95.0	1.45

A. Noise Robustness

When evaluated under moderate background noise conditions (10–20 dB SNR), average STT accuracy decreased to 87.0% and intent accuracy to 90.2%. However, under moderate background noise conditions, speech recognition accuracy decreased slightly, which is expected for browser-based speech recognition systems operating in noisy environments.

B. User Study

A usability study involving 12 participants (ages 18–35) assessed interaction comfort and perceived responsiveness. On a 5-point Likert scale, the system received an average ease-of-use score of 4.2, indicating that most users found voice-based control natural and intuitive.

C. Time Comparison

A comparison of task execution time demonstrated that SAYLINK reduces user effort. Manual interaction using mouse and typing averaged 2.4 seconds per action, whereas voice-based interaction averaged 1.45 seconds, resulting in a ~39% time reduction. This highlights the efficiency advantage of hands-free interaction for continuous social media browsing.

XII. DISCUSSION

SAYLINK demonstrates that voice-based interaction can meaningfully enhance usability in social media environments, particularly for scenarios where hands-free control is desirable. The system enables users to perform core actions such as liking, commenting, searching, and navigating without relying on manual input, resulting in faster task execution and increased accessibility. The modular architecture further allows the speech recognition and intent parsing components to be upgraded independently, making the system extensible for future improvements, such as multilingual support or large-model NLU integration.
However, several limitations remain. The performance of browser-based STT varies across devices and browsers, which can lead to inconsistent transcription quality. Additionally, commands involving relational or contextual references (e.g., “like this post”) require heuristics to determine the target, and such heuristics may fail in ambiguous cases. Another limitation concerns privacy, particularly in environments with background noise, where speech recognition performance may vary across different browsers and hardware configurations, which may affect recognition accuracy across different environments.

A. Advantages

1) Enables significantly faster interactions for routine tasks.
2) Enhances accessibility for visually impaired and mobility-impaired users.
3) Modular architecture allows seamless upgrades of STT or NLU components.

B. Limitations

1) Browser STT accuracy varies with hardware and platform differences.
2) Context-dependent phrases require heuristic interpretation and may be error-prone.
3) Speech recognition accuracy may decrease in noisy environments or with varied accents.

XIII. COMPARATIVE ANALYSIS

Traditional social media platforms rely heavily on manual interaction methods such as clicking, scrolling, and typing, which can be slow and restrictive for certain users. In contrast, existing voice-enabled interfaces (such as built-in voice search features or external assistants like Siri and Google Assistant) typically only support single-turn commands or limited query functions, and they do not perform end-to-end in-app social actions. SAYLINK addresses this gap through continuous, multi-intent voice interaction that directly maps spoken commands to platform operations, enabling actions such as liking, commenting, navigating feeds, following users, and opening posts.

Furthermore, unlike external voice assistants that require additional integration layers, SAYLINK performs intent recognition and UI state updates internally, ensuring smoother interaction flow. The use of browser-based speech recognition ensures low latency and seamless real-time interaction without requiring external processing, which further enhances flexibility and robustness. As a result, SAYLINK provides a more context-aware, hands-free, and seamless social interaction experience compared to manual UI navigation or voice-search-only solutions.

XIV. APPLICATIONS AND EXTENSION POTENTIAL

The architectural modularity of SAYLINK allows it to be extended into several domains and enhanced with future capabilities.

A. Multilingual STT and NLU

Integrating multilingual STT models (e.g., advanced multilingual speech recognition models) would expand accessibility to non-English speakers and users with diverse accents, making the system more inclusive and globally adaptable.

B. Voice and Biometric for an Authentication

Voice identification techniques can be introduced as an additional authentication layer, ensuring secure login or approval of sensitive operations without requiring manual password input.

C. Personalized Content and Recommendations

Interaction patterns from voice commands (e.g., frequently liked topics, repeated search terms) can be processed by recommendation algorithms to tailor content feeds, improving user engagement and relevance.

D. Mobile and AR/VR Interfaces

The platform can be deployed in mobile environments or immersive AR/VR contexts, enabling hands-free control in scenarios where physical interaction is inconvenient—such as during exercise, cooking, outdoor navigation, or accessibility-assisted usage.

Overall, SAYLINK presents a framework that can evolve beyond web-based social media to become a generalizable voice interaction layer for digital platforms requiring natural, accessible, and efficient user interfaces.

XIII. CONCLUSION

SAYLINK demonstrates that voice-based interaction can significantly enhance the usability and accessibility of social media platforms. By integrating Speech-to-Text, Natural Language Understanding, and Text-to-Speech within a modular full-stack architecture, the system enables hands-free execution of core social media actions such as liking, commenting, searching, and navigation. Experimental evaluation indicates notable improvements in task efficiency and user comfort, along with strong intent recognition accuracy under standard conditions. User feedback further supports that voice interaction can reduce operational effort and benefit users with motor or visual impairments. Overall, SAYLINK shows that conversational interaction can serve as a viable alternative to manual input in dynamic social environments.

XIV. FUTURE WORK

While the current SAYLINK system demonstrates effective voice-based interaction, future work will focus on enhancing the platform using the existing technology stack. Multilingual voice command support will be added to improve accessibility for diverse users. Context-aware command handling will also be introduced to support more natural interactions such as referencing previously viewed posts. These improvements will be implemented through enhanced command processing and better state management within the React-based frontend.

Additionally, real-time responsiveness and scalability will be improved using Socket.IO and optimized Node.js APIs. Security enhancements such as secure communication and improved data handling will also be incorporated. The system will further be optimized for different devices and responsive layouts, enabling smoother interaction across platforms. These enhancements aim to evolve SAYLINK into a more robust and intelligent voice-driven social media application.

REFERENCES

[1] César González-Mora, Irene Garrigós, Marco A. Celdrán-Bernabeu, David Giner-Sánchez,Jose-Norberto Mazón, "Augmenting Tourism Websites with Voice Commands," *Institute of Electrical and Electronics Engineers (IEEE)*, 2025, DOI-10.1109.

[2] Libo Joel A., Dr. Christina Magneta, "Voice-Enabled Inclusive E-Commerce Platform," *International Journal for Research in Applied Science & Engineering Technology (IJRASET)* ISSN: 2321-9653; IC Value: 45.98, Volume 13 Issue I Jan 2025.

[3] Anshul Gupta, Nileshkumar Patel, Shabana Khan, "Automatic Speech Recognition Technique for Voice Command," *International Conference on Intelligent Communication and Computational Techniques (IEEE-32331)*, 2023.

[4] Antonia Barbaric, Cosmin Munteau, Heather J. Ross, "Design, Development, and Usability Evaluation of Medly Voice App," *D.O.I:10.1101*.

[5] Gitansh Kumar, Manmeet Kaur Matharu, Gaurank Joshi, Jatin Chauhan, Ankit Kumar, "Voice-Assisted E-Commerce Website Using Django and GTTS," *International Journal of Advanced Research in Computer Science*, 2024.

[6] Mandeep Singh Kandhari, Farhana Zulkernine, Haruna Isah, "Cloud-Based Voice Controlled E-Commerce Web Application," *International Journal of Cloud Computing and Intelligent Systems*, 2024.