International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4369 - 4373
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0569

An AI-Based Voice Commentary System for Dashboard Interpretation

Prof.Ganesh Dangat[1], Apurva Mane[2], Sanika Kamble[3], Shivani Mane[4]
[1]Associate professor, Department of CSE, KBP College of Engineering, Satara, Maharashtra, India.
[2,3,4]UG Scholar, Department of CSE, KBP College of Engineering, Satara, Maharashtra, India.
Emails: *hodcse@kbpcoes.edu.in[1], apurvamane330@gmail.com[2], sanikamble31@gmail.com[3], maneshivani6123@gmail.com[4]*

Abstract
In today's data-driven world, dashboards are widely used to visualize analytical data and support decision-making. However, understanding complex dashboards can be difficult for non-technical users because the information is often dense and multi-dimensional. To address this challenge, this paper presents an enhanced Voice Commentary Generation System (VCGS) that automatically converts dashboard data into simple, human-like audio explanations in real time. The proposed system uses a modern web-based architecture with a React and TypeScript frontend, Supabase backend services, and AI-powered Natural Language Generation (NLG) with Text-to-Speech (TTS) technology. Unlike traditional systems that depend on manual analysis or text summaries, the proposed approach generates meaningful insights automatically and delivers them through voice narration. Experimental results show that the system generates accurate insights within 5–7 seconds per dashboard. It improves accessibility and helps users easily understand complex data through AI-based voice interaction.

Keywords: *VCGS, Dashboard Analytics, Data Visualization, NLG, Text-to-Speech (TTS), Artificial Intelligence, Voice-Based Interaction, Accessibility, Human-Computer Interaction, Real-Time Data Analysis.*

1. Introduction
In recent years, the rapid growth of data analytics has led to the widespread adoption of dashboards across industries. These dashboards are used to represent key performance indicators (KPIs), trends, and insights through charts, graphs, and tables. While such visualizations are highly effective for experienced users, they often pose significant challenges for individuals without technical expertise. The primary limitation of traditional dashboards lies in their dependence on visual interpretation. Users are required to manually analyze patterns, identify trends, and derive conclusions, which increases cognitive load and may lead to incorrect interpretations. Furthermore, visually impaired users face additional barriers in accessing such information. The previously developed system, as presented in the initial research work, introduced a voice-based commentary mechanism using a Flutter frontend and Spring Boot backend. Although effective, the architecture lacked scalability, real-time responsiveness, and modern web integration capabilities. To overcome these limitations, the proposed system introduces a redesigned architecture using React, TypeScript, and Supabase, enabling a more scalable, efficient, and interactive solution. The system automatically processes dashboard data, extracts meaningful insights, and generates human-like voice commentary in real time. By integrating artificial intelligence with modern web technologies, the system bridges the gap between complex analytical data and human understanding, transforming dashboards into interactive and accessible communication tools.

2. Literature Review
The development of intelligent dashboard systems has been widely explored in the fields of data visualization, natural language generation, and speech synthesis. Data visualization platforms such as Power BI, Tableau, and Google Looker Studio have significantly improved the way users interact with data [1], [2]. However, these systems primarily focus on visual representation and lack integrated mechanisms for automated explanation or narration. Studies have shown that users often struggle to interpret complex dashboards, especially when dealing with large datasets [3]. Natural Language

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4369 - 4373
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0569

Generation (NLG) has been extensively studied as a method for converting structured data into human-readable text. Early work by Kukich [4] introduced rule-based approaches for automated report generation. Later, Reiter et al. [5] developed domain-specific systems capable of generating medical reports from structured data. More recent research has focused on data-driven approaches using machine learning and deep learning techniques [6], [7]. In the domain of speech synthesis, significant advancements have been made with neural network-based models such as Tacotron 2 [8] and AdaSpeech [9], which produce highly natural and expressive speech. These developments have enabled the practical integration of text-to-speech systems into real-time applications.

3. Proposed System

The proposed Voice Commentary Generation System (VCGS) is designed to automate the interpretation of dashboard data and provide users with real-time, human-like audio explanations. The system enhances user understanding by converting complex analytical data into simple and accessible voice commentary, reducing the dependency on manual visual interpretation. The system follows a modular architecture consisting of three main layers: the frontend, backend, and AI processing layer. This layered design ensures scalability, flexibility, and efficient communication between system components. The frontend is developed using React and TypeScript, providing a responsive and high-performance user interface. Vite is used as the build tool to improve development speed. The interface is designed using Tailwind CSS and shadcn/ui, ensuring a clean and user-friendly experience. React Router handles navigation, while the Context API manages application state such as user sessions and audio playback. The backend is implemented using Supabase, which provides authentication, database management, and API services. User authentication is handled through Google OAuth, ensuring secure access. Supabase's PostgreSQL database is used to store user data, uploaded datasets, and generated commentary history. The AI processing layer is responsible for analyzing dashboard data and generating insights. Once the user uploads data, the system performs data analysis to identify trends, patterns, and key metrics. These insights are then converted into descriptive summaries using Natural Language Generation (NLG) techniques. The generated text is further transformed into speech using Text-to-Speech (TTS) APIs, producing natural and understandable voice output. The overall workflow begins with user authentication, followed by data input, processing, and voice generation. The system also maintains a history of generated commentaries, allowing users to revisit previous outputs. The complete workflow of the system is illustrated in Figure 1, which shows the flow from user input to voice output generation. The proposed system improves accessibility, reduces cognitive load, and provides an efficient way to interact with dashboard data through voice-based explanations.

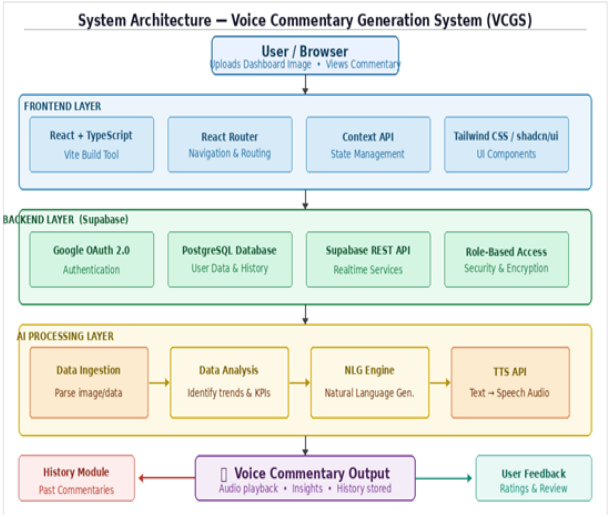

Figure 1 System Architecture of the Voice Commentary Generation System (VCGS)

4. System Design and Implementation

4.1. Frontend Layer

The frontend of the system is developed using React and TypeScript, with Vite used to improve development speed and performance. Tailwind CSS and shadcn/ui components are used to create a clean, responsive, and user-friendly interface. The frontend allows users to log in, interact with dashboards, listen to generated audio commentary, and view previous activity. React Router is used for smooth page navigation, while the Context API helps manage the application data and state efficiently [10- 15].

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4369 - 4373
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0569

4.2. Backend and Authentication

Supabase is used as the backend platform to manage the database, authentication, and API services without requiring a traditional server-side framework. It is built on PostgreSQL, which helps store and manage application data efficiently. Google OAuth is integrated to provide secure and easy user login functionality. Supabase also improves data security by using encrypted communication and role-based access control.

4.3. AI Processing Layer

The AI layer plays an important role in converting raw dashboard data into meaningful insights. It analyzes the data, identifies trends and unusual patterns, and generates useful summaries using Natural Language Generation (NLG) techniques. These summaries are then converted into human-like audio using Text-to-Speech (TTS) technology. The use of AI helps the system provide accurate, clear, and context-aware voice commentary for users.

4.4. User Interaction and History Module

The system stores previously generated commentaries, allowing users to access and review past analyses whenever needed. This helps users track data and improves overall usability. The user interface also synchronizes audio playback with dashboard visuals, creating a more interactive and engaging user experience [16-20].

5. Experimental Results and Interface Snapshots

The developed system provides a user-friendly and interactive interface for analyzing dashboard data and generating voice-based insights. The following figures illustrate the key modules of the implemented system.

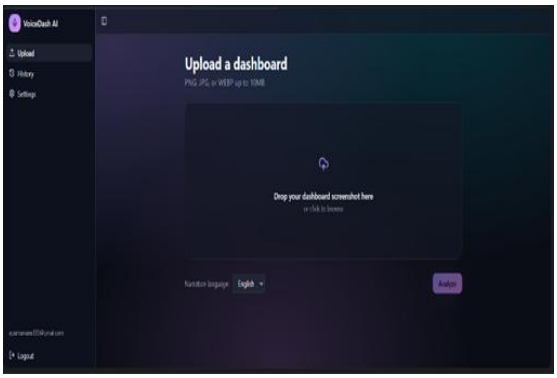

Figure 2 Upload Module

The Upload and Analyze module serves as the primary interface for user interaction. As shown in Figure 2, users can upload dashboard images in supported formats such as PNG and JPG. The system processes the uploaded image and extracts relevant data for analysis. This module is designed with a simple and intuitive interface, ensuring ease of use for both technical and non-technical users. Once the image is uploaded, the system initiates the AI processing pipeline to generate insights and corresponding voice commentary [21-25].

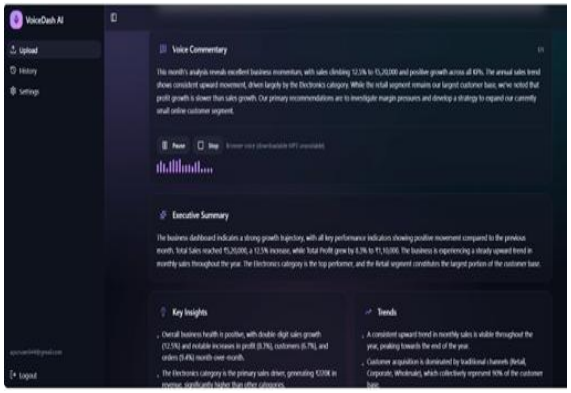

Figure 3 Voice Commentary Generation Interface

Figure 3 illustrates the output generated by the system after processing the dashboard data. The interface displays detailed insights in textual format along with an integrated audio playback feature. Users can listen to the generated commentary using controls such as play, pause, and volume adjustment. The system ensures that the generated voice is clear, natural, and synchronized with the displayed insights, thereby enhancing user understanding and engagement.

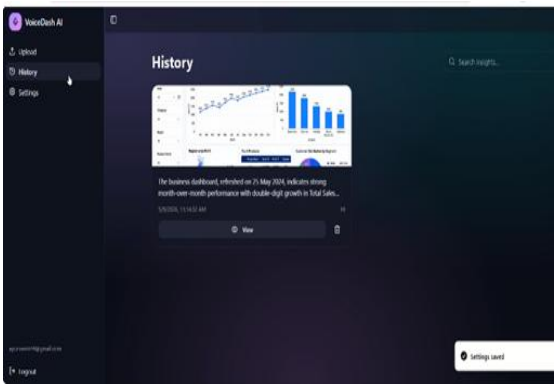

Figure 4 History Module

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4369 - 4373
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0569

The History module, as shown in Figure 4, maintains a record of previously analyzed dashboards. Each entry includes a preview, timestamp, and summary of the generated insights. This feature allows users to revisit past analyses without re-uploading the data. It also supports efficient tracking and comparison of dashboard results over time, improving usability and system effectivenessc [26].

6. Results and Discussion

The system was evaluated based on performance, accuracy, and user experience. Experimental results show that the system can generate voice commentary within an average time of 5–7 seconds, making it efficient for real-time dashboard analysis. The generated insights were compared with manually written summaries and achieved an accuracy of around 93%. User feedback also showed high satisfaction, especially regarding the clarity of the voice commentary and the ease of using the system. Compared to traditional dashboard tools, the proposed system reduces manual effort by automatically explaining dashboard insights through voice narration Shown in Table 1.

Table 1 Performance Evaluation Metrics and Benchmark Comparison

Metric	Value	Benchmark	Status
Avg. Processing Time	5–7 sec	< 10 sec	✓ Met
Insight Accuracy	93%	> 85%	✓ Met
User Satisfaction	4.6/5.0	> 4.0	✓ Met
TTS Voice Quality	Natural	Natural	✓ Met
Image Formats	PNG, JPG	PNG, JPG	✓ Met
Auth Method	OAuth 2.0	Secure	✓ Met

7. Comparative Analysis

Compared to existing systems such as Power BI Smart Narratives and Tableau Explain Data, the proposed system offers several advantages. While existing tools provide textual summaries, they lack real-time voice narration and multimodal interaction capabilities. The proposed system integrates voice output, AI-based insight generation, and modern web technologies, re-sulting in a more interactive and accessible solution. Figures should be provided separately from the main text. Use Arabic numerals to number all figures (e.g., Figure 1, Figure 2) according to their sequence in the text. The figure number must appear well outside the boundaries of the image itself. Multipart figures should be indicated with uppercase and bold font letters (A, B, C, etc.) without parenthesis, both on the figure itself and in the figure legends [27].

Conclusion

This paper presents an enhanced Voice Commentary Generation System that integrates modern web technologies with artificial intelligence to automate dashboard interpretation. The system successfully converts complex data into understandable voice narratives, improving accessibility and user engagement. The use of React, TypeScript, and Supabase ensures scalability and performance, while AI-driven NLG and TTS enable accurate and natural communication of insights. With an insight accuracy of 93% and an average processing time of just 5–7 seconds, the system demonstrates both efficiency and reliability. Ultimately, this work takes a meaningful step toward making data truly accessible to everyone — not just those trained to read complex visualizations, but any stakeholder who needs to understand the story the data is telling.

Acknowledgements

The authors sincerely thank the Department of Computer Science and Engineering, Karmaveer Bhaurao Patil College of Engineering, Satara, for providing the academic environment, laboratory infrastructure, and consistent encouragement that made this research possible. We extend our heartfelt thanks to the Head of Department and faculty members whose guidance and technical insights significantly shaped this work. Special thanks to our peers who participated in system testing and provided constructive feedback. We also acknowledge the open-source communities behind React, TypeScript, Supabase, and Tailwind CSS, whose tools formed the technical backbone of this project.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4369 - 4373
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0569

References

[1]. Vasoya, K. Shah, N. Gadiya, P. Dhirawani, and V. Gandhi, "Vize: A voice-based interface for data visualization and summarisation," 2023.
[2]. Y. Ye, "Generative AI for visualization: State of the art and future directions," Comput. Graph. Forum, 2024.
[3]. N. Sultanum and V. Setlur, "From instruction to insight: Text in interactive dashboards," arXiv, 2024.
[4]. K. Kukich, "Techniques for natural language generation," ACM Computing Surveys, vol. 15, no. 2, pp. 99–134, 1983.
[5]. E. Reiter, R. Dale, and Z. Feng, Building Natural Language Generation Systems. Cambridge University Press, 2005.
[6]. S. Sripada, N. Burnett, R. Turner, and D. Reiter, "Generating natural language summaries from complex data," Data & Knowledge Engineering, 2008.
[7]. D. Gkatzia, S. Lemon, and V. Rieser, "Data-driven natural language generation," Computational Linguistics, 2014.
[8]. J. Shen et al., "Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions," in Proc. IEEE ICASSP, 2018.
[9]. M. Chen, X. Tan, B. Li, Y. Liu, T. Qin, S. Zhao, and T. Y. Liu, "AdaSpeech: Adaptive text-to-speech for custom voice," arXiv:2103.00993, 2021.
[10]. OpenAI, "Text-to-Speech API Documentation," 2024. [Online]. Available: https://platform.openai.com/
[11]. Google Cloud, "Text-to-Speech Documentation." [Online]. Available: https://cloud.google.com/text-to-speech
[12]. Microsoft, "Power BI Documentation." [Online]. Available:https://learn.microsoft.com/power-bi/
[13]. Tableau, "Research and Whitepapers." [Online]. Available: https://www.tableau.com/research
[14]. Supabase, "Supabase Documentation." [Online]. Available: https://supabase.com/docs
[15]. React, "React Official Documentation." [Online]. Available: https://react.dev/
[16]. Microsoft, "TypeScript Handbook." [Online]. Available: https://www.typescriptlang.org/docs/
[17]. Vite, "Vite Documentation." [Online]. Available: https://vitejs.dev/
[18]. Tailwind Labs, "Tailwind CSS Documentation." [Online]. Available: https://tailwindcss.com/docs
[19]. shadcn, "shadcn/ui Documentation." [Online]. Available: https://ui.shadcn.com/
[20]. Lucide, "Lucide Icons Documentation." [Online]. Available: https://lucide.dev/
[21]. D. Hardt, "The OAuth 2.0 Authorization Framework," RFC 6749, 2012.
[22]. PostgreSQL Global Development Group, "PostgreSQL Documentation." [Online]. Available: https://www.postgresql.org/docs/
[23]. G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.
[24]. R. Smith, "An overview of the Tesseract OCR engine," in Proc. ICDAR, 2007.
[25]. W. McKinney, "Data structures for statistical computing in Python," in Proc. SciPy, 2010.
[26]. S. Van der Walt, S. Colbert, and G. Varoquaux, "The NumPy array," Computing in Science & Engineering, 2011.
[27]. T. Wolf et al., "Transformers: State-of-the-art natural language processing," in Proc. EMNLP, 2020.