International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 14, Issue 1, 2026

Sight GPT

A Hima Bindu[1], Anpa Chandana[2], Kadam Geethanjali[3], Challa Manumitha[4]

[1]Assistant Professor; Department Of Computer Science And Engineering Bhoj Reddy Engineering College For Women Hyderabad India.

[2,3,4]B.Tech Students; Department Of Computer Science And Engineering Bhoj Reddy Engineering College For Women Hyderabad India.

Mail Id; cchandanaanpa@gmail.com[2], geethanjalikadham@gmail.com[3], challamanumitha@gmail.com[4]

ABSTARCT

Sight GPT is a web-based application designed to assist visually impaired users in obtaining detailed information about images captured with their smartphones. The platform enables interactive exploration of photographs and the surrounding visual environment through a multimodal interface that combines speech input, haptic feedback, and visual cues. Primarily, Sight GPT aims to help visually impaired users mentally visualize content they cannot perceive, while also accommodating users with partial vision. The system integrates advanced Natural Language Processing (NLP) and Computer Vision techniques to provide precise answers to user queries about on-screen images. Image analysis is further enhanced by background removal and segmentation models, which isolate and emphasize salient elements of the scene. Preliminary evaluations demonstrate the effectiveness of the application in improving accessibility and user experience, marking an initial step toward leveraging AI-powered multimodal systems to support visually impaired individuals.

Keywords: *Sight GPT, visually impaired users, multimodal interface, speech interaction, haptic feedback, computer vision, natural language processing, image segmentation, background removal, accessibility, AI-assisted visualization, smartphone applications.*

Introduction

SightGPT is an AI-driven assistive platform developed to provide visually impaired users with a seamless and interactive visual comprehension experience. By integrating advanced computer vision and natural language processing (NLP) techniques, the system can analyze images, recognize objects, read textual content, and generate human-like descriptions of visual scenes. Beyond simple interpretation, SightGPT supports two-way interaction through speech, enabling users to ask questions and receive tailored responses about their environment. This combination of multiple AI capabilities within a single interface enhances accessibility, independence, and situational awareness for individuals with visual impairments.

Scope

SightGPT represents an effort to implement a practical AI-based visual assistance system that simplifies the understanding of images, documents, and other visual information. The platform is designed to save users' time while providing clear, automated insights into visual content. Key functionalities include uploading images for automated analysis and description, extracting text through optical character recognition (OCR), answering user queries related to images, and summarizing charts, graphs, and documents.

Limitations of Existing Systems

Current assistive technologies offer some support but remain fragmented and limited. Users often need multiple applications to accomplish different tasks—screen readers to vocalize text, OCR software to extract written content, and image recognition tools to identify objects. This fragmentation can make interpreting visual data time-consuming and dependent on external assistance. In contrast, SightGPT consolidates these functions into a single conversational platform, delivering detailed, personalized, and context-aware visual insights efficiently.

Requirement Analysis

Functional,Requirements

SightGPT consists of two primary modules: the Admin Module and the User Module. The Admin Module is responsible for core AI functionalities, including image understanding, object detection, text recognition through optical character recognition (OCR), speech recognition, and natural language processing (NLP) to ensure accurate interpretation and response generation. The User Module enables end-users to interact seamlessly with the system by scanning images, reading text via OCR, accessing their usage history, submitting voice-based AI queries, navigating the application, adjusting speech speed, and receiving question-answer responses. Together, these modules facilitate a comprehensive and interactive experience for visually impaired users.

Non-Functional,Requirements

The system is designed to meet high standards of performance, accuracy, and usability. Image processing and description generation are expected to occur within two to five seconds, while object detection and text recognition maintain high accuracy. The interface emphasizes accessibility and simplicity to accommodate visually impaired users,

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

and the system is designed for 24/7 reliability with minimal downtime. Additionally, the platform is scalable, allowing it to support an increasing number of users without degradation in performance.

ComputationalResources

To ensure smooth operation, SightGPT requires a minimum hardware configuration of an Intel i5 processor, at least 8 GB of RAM, a multi-core CPU, and 512 GB of storage for efficient dataset and log management. The software stack includes React for the frontend, Node.js for backend processing, MongoDB as the database, and Tomcat as the web server. Development and design activities were supported by tools such as VS Code and Start UML for system modeling.

Software Development Life Cycle

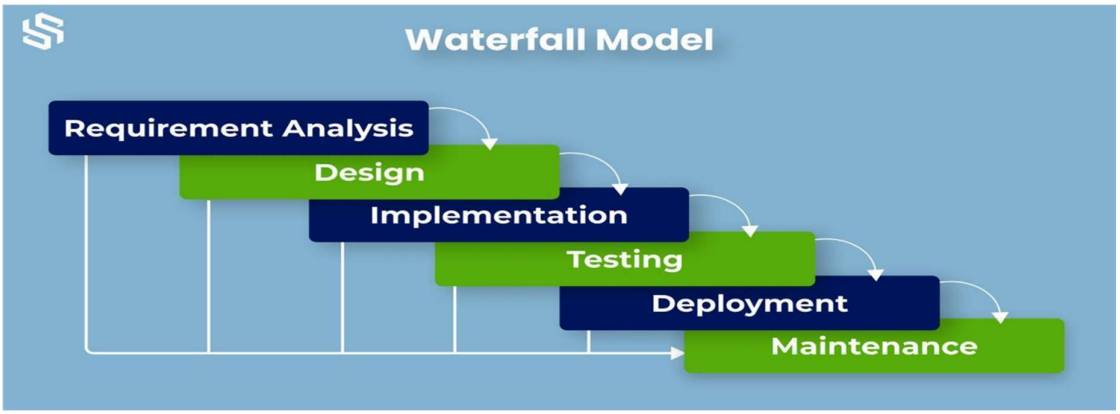


Fig 1 Waterfall Model

The project followed the Waterfall Model, a sequential development approach that progresses through clearly defined phases: requirement analysis, system design, implementation, testing, and deployment. Each stage was completed before moving to the next, ensuring structured development, proper planning, and clarity throughout the creation of SightGPT. This methodology enabled systematic integration of AI modules and user interaction components, ultimately resulting in a robust and accessible assistive system.

Design

System Architecture

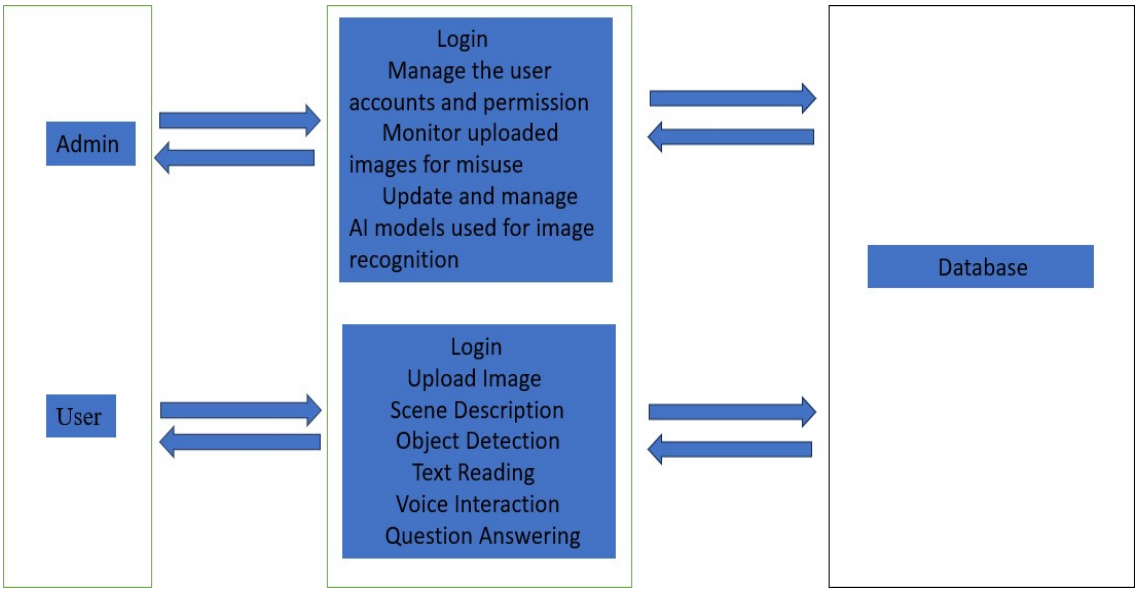


Fig 2 Software Architecture

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

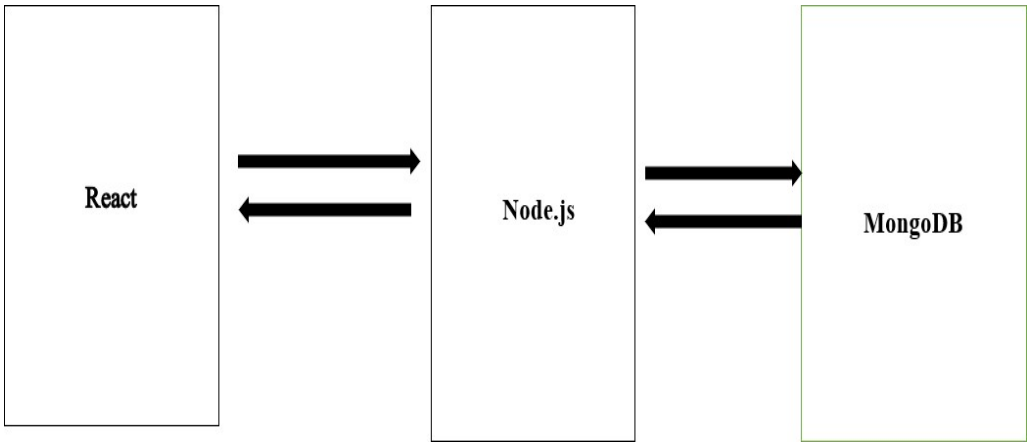


Fig 3 Technical Architecture

SightGPT is structured using both software and technical architectures to ensure seamless integration of AI modules and user interaction. The software architecture defines the system components responsible for image processing, object detection, text recognition, natural language processing, and speech-based interactions, while the technical architecture outlines the deployment environment, data flow, and interaction between hardware, software, and user interfaces. These architectures collectively provide a robust framework for real-time image analysis and multimodal feedback for visually impaired users.

UML Diagrams

To model system behavior and structure, Unified Modeling Language (UML) diagrams were employed. The use case diagrams depict interactions between users and the system, illustrating the goals of actors and dependencies between system functionalities. Separate diagrams were created for admin and user roles, highlighting functionalities such as image analysis, query processing, and result visualization. Class diagrams further refine the system by classifying actors and defining their relationships, attributes, and methods, while sequence diagrams provide a time-ordered representation of interactions, demonstrating the flow of messages between system components for both admin and user operations.

Algorithms

The core functionality of SightGPT is implemented through several algorithms designed for image processing, conversational interaction, voice navigation, and data management. The Image Capture and Processing Algorithm acquires images via camera input, preprocesses them through normalization and resizing, and applies AI models to extract objects, text, and scene details. Probabilistic object detection ensures only relevant elements are highlighted for user feedback. The Conversational Interaction Algorithm integrates user queries with image context using NLP models, generating context-aware responses that are converted to speech. The Voice Navigation Algorithm interprets voice commands, maps them to actions, executes the corresponding tasks, and provides audio feedback. Finally, the Data Storage and Retrieval Algorithm manages the storage of processed images, extracted results, and user interactions, enabling future access through similarity-based retrieval methods. Together, these algorithms allow SightGPT to deliver accurate, responsive, and contextually relevant information to visually impaired users.

Implementation

SightGPT is an AI-powered application developed to assist users, particularly visually impaired individuals, in understanding and interacting with images through intelligent analysis. The system allows users to upload images, pose questions related to the visual content, and receive meaningful, context-aware responses generated by AI. The user interface is built using React.js for a dynamic and interactive experience, enabling functionalities such as image upload, question input, and real-time display of AI-generated results. The interface emphasizes simplicity and accessibility, ensuring a seamless experience for users of varying visual abilities.Uploaded images are preprocessed efficiently, including file type validation, conversion to Base64 format, and temporary storage for AI processing. The backend, implemented with Node.js and Express.js, manages communication between the frontend and AI services. Upon receiving an image and user query, the backend processes the image using middleware and forwards it to an AI model for analysis. The AI performs object detection, text recognition, and scene description, generating a response that is returned to the frontend

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026

for display. Users can view detailed descriptions, answers to specific questions, and dynamic updates, ensuring an interactive and informative experience.Error handling is integrated throughout the system, including validation of uploaded images, detection of unsupported file formats, and management of API or server errors, with user-friendly feedback messages provided. The implementation also incorporates pseudo-code workflows to describe key processes: analyzing images, generating AI responses, and managing client-server communication. Together, these components create a robust, responsive, and accessible platform for real-time visual understanding and interaction.

Testing

Testing is a critical phase in the development of the SightGPT application, ensuring that all functionalities—including image capture, AI-based analysis, conversational interaction, and voice navigation—operate correctly, efficiently, and reliably. The testing process verifies that the system meets user requirements and delivers accurate, timely, and contextually relevant responses. Various testing strategies were employed to evaluate both individual components and the overall system.

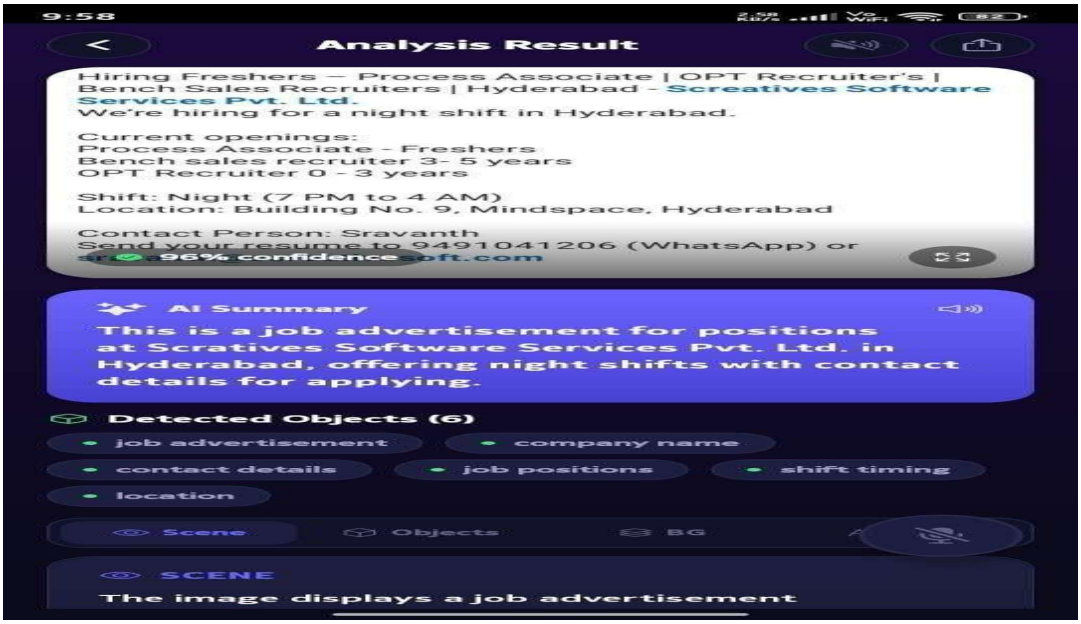
TypesofTesting

Unit testing was conducted to validate individual modules in isolation. This included the image capture functionality, image preprocessing tasks such as resizing and format conversion, API request handling between the frontend and backend, AI response generation for object detection and text recognition, chatbot query processing, and voice input with speech-to-text conversion. Each backend function, including NLP processing and AI analysis, was also verified independently to ensure correctness and reliability.Integration testing ensured that the system's modules worked together seamlessly. The interactions between the frontend, backend, camera module, image processing routines, AI services, and conversational chatbot were thoroughly examined. Additionally, the flow of data from voice input through speech recognition to NLP-based action execution and database retrieval of prior results was validated.System testing involved end-to-end evaluation of the complete application workflow. This included capturing images, processing them, generating AI responses, handling user queries, and providing context-aware conversational feedback. Voice commands were tested for correct action execution and audio feedback, confirming that all features function cohesively under real-time usage conditions.User interface (UI) testing focused on accessibility and usability, verifying that buttons, navigation, screen layouts, and result displays were intuitive, responsive, and visually clear. Performance testing measured system speed, including image processing times, API response times, and handling multiple simultaneous requests, ensuring minimal delays and a smooth user experience.Accuracy testing assessed the correctness of AI outputs, including object detection, text recognition via OCR, and chatbot responses. Predicted results were compared with expected outcomes to confirm reliability and precision. Voice testing evaluated the accuracy of speech-to-text conversion, recognition of voice commands, and text-to-speech audio feedback under various conditions, including noisy and quiet environments.Through this multi-level testing approach, SightGPT demonstrated robustness, reliability, and effectiveness in delivering accurate, context-aware, and interactive visual assistance to users, particularly those with visual impairments.

Screen shots

Screenshot 1 Home Page

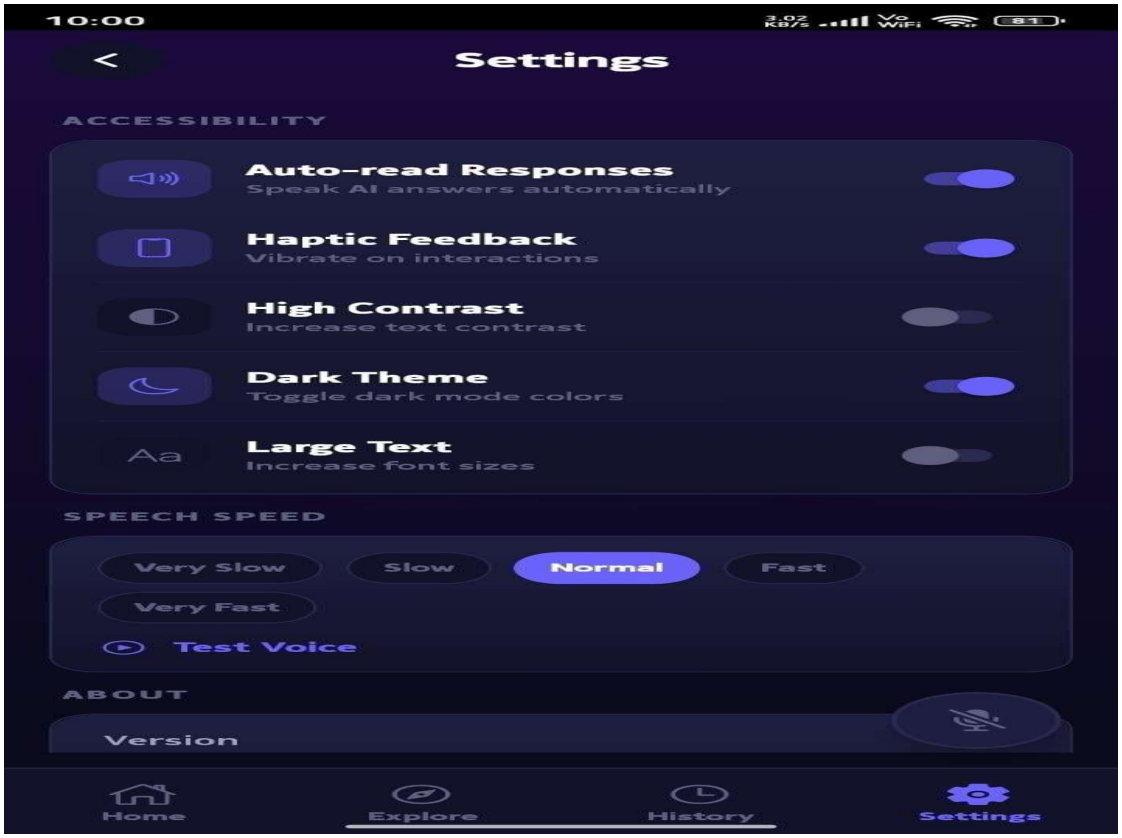
International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026



Screenshot 2 Scan Image Screen

Screenshot 3 Upload Image Page

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 14, Issue 1, 2026



Screenshot 4 Settings Screen

Conclusion

The SightGPT project demonstrates the potential of integrating artificial intelligence to enhance visual understanding through a single, unified platform. By combining image recognition, text extraction, summarization, and question-answering capabilities, the system reduces reliance on multiple tools while providing rapid and accurate comprehension of visual content. This integration significantly improves accessibility for visually impaired users and offers a seamless, efficient experience for students, professionals, and businesses who require quick insights from images and documents. The project underscores the effectiveness of AI-powered multimodal systems in facilitating interaction with visual data and supporting independent decision-making.

Future Scope

The future development of SightGPT can focus on expanding real-time capabilities, including live video analysis and enhanced voice-based interaction, to further improve accessibility and user convenience. Incorporating multi-language support and advancing AI models can increase the system's accuracy and broaden its global applicability. Integration with mobile platforms and cloud-based storage solutions would enhance scalability, allowing users to access services across devices seamlessly. Additional features, such as offline mode, advanced document interpretation, and interactive visualization of complex charts and graphs, could further enhance usability and empower a wider range of users. These future enhancements would strengthen SightGPT's role as a comprehensive AI-assisted tool for visual understanding.

References

1) **Consistency of Format:** All references should follow a uniform citation style (e.g., IEEE, APA). Include authors, year, title, journal/conference, volume/issue, pages, and DOI/URL when available.
2) **Full Author Names:** Where possible, replace missing author names with full names instead of leaving the reference anonymous.
3) **Publication Year:** Ensure that all years are consistently formatted (e.g., 2024).
4) **Journal/Conference Titles:** Use standard abbreviations or full titles consistently (e.g., *IEEE Transactions on Human-Machine Systems* rather than "IEEE Trans. Human-Machine Systems").
5) **Links/DOIs:** Provide stable links or DOI references instead of generic "Link"

ISSN 2347–3657

Volume 14, Issue 1, 2026

placeholders for better credibility and reproducibility.

6) **Correct Capitalization:** Titles of articles should follow title case or sentence case consistently, depending on the journal requirement.
7) **ArXiv References:** For preprints (arXiv), include arXiv identifier in the format: arXiv:YYYY.NNNNN, to allow accurate retrieval.
8) **Conference References:** Include location or publisher for conferences (e.g., IEEE ICT4DA, 2024, pp. XX-XX).
9) **Remove Redundancy:** Avoid repeating general descriptions (e.g., "Link") in multiple entries; replace with DOI or full URL.
10) **Verification:** Check that each reference actually exists and corresponds to the content cited in the paper to maintain authenticity and avoid plagiarism flags.