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

Sign-Language Media Hub

K Shireesha[1],R Bhanupriya[2],T Likitha[3],P Manogna[4]
[1]Associate 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.
rbhanupriya2907@gmail.com, likithathammali7819@gmail.com, manognareddypanga@gmail.com

Abstract
The Sign-Language Media Hub is a video streaming platform designed to improve digital accessibility for individuals who are deaf or hard of hearing. The system enables content creators to upload videos with captions, which are then processed using external application programming interfaces (APIs) to generate synchronized sign-language interpretation. The platform emphasizes inclusivity, usability, and scalability by incorporating features such as side-by-side video playback, adjustable playback speed, and customizable delay controls. These capabilities aim to enhance comprehension and provide a flexible viewing experience for users with hearing impairments. Additionally, the platform simplifies the creation of accessible multimedia content for content creators while maintaining efficient processing and delivery. With a scalable architecture and user-centered design, the Sign-Language Media Hub contributes to advancing inclusive digital media and improving accessibility in online video platforms.

Keywords: *Accessibility, Sign Language, Video Streaming, Deaf Community, Inclusive Technology, Media Platform, Assistive Technology*

Introduction
Digital media has become the primary medium for education, entertainment, and communication. However, accessibility barriers continue to limit equal participation for deaf and hard-of-hearing individuals. While captions provide partial accessibility, they do not fully address the communication needs of users who rely primarily on sign language. Sign-Language Media Hub is designed as an accessible video streaming platform that integrates sign language interpretation into digital video content. The system enables users to upload videos along with captions, which are then processed to generate synchronized sign language displays. By incorporating automated interpretation and flexible playback controls, the platform aims to improve comprehension, promote inclusivity, and expand access to digital content for a broader audience.

Existing System
Current video streaming platforms primarily depend on subtitles or captions to support deaf and hard-of-hearing users. Although captions provide textual representation of spoken content, they often fail to convey tone, context, and linguistic nuances that are naturally expressed in sign language. Additionally, caption accuracy may vary significantly, particularly when auto-generated captions are used. Some educational institutions and media organizations manually embed sign language interpreters within videos; however, this approach requires additional production effort, increases costs, and lacks scalability. Furthermore, most existing platforms do not provide built-in mechanisms for automated sign language integration, limiting accessibility improvements across diverse content libraries.

Problems in Existing System
The limitations of current systems create multiple accessibility challenges. First, caption-only accessibility does not adequately serve users who rely primarily on sign language, and automatically generated captions may contain inaccuracies that affect comprehension. Second, manual embedding of sign language interpreters is resource-intensive and impractical for large-scale content production. Third, the absence of centralized sign language integration across video platforms prevents consistent accessibility standards. Finally, limited support for regional and international sign languages restricts usability for global audiences, reducing the effectiveness of accessibility solutions.

Proposed System
The proposed Sign-Language Media Hub addresses these limitations by incorporating automated sign language interpretation into video content. The system processes uploaded captions and converts them into synchronized sign language displays using external APIs and caption processing mechanisms. The platform supports dual video playback, allowing users to view both the original content and sign language interpretation simultaneously. Additional features such as playback speed adjustment, customizable synchronization delay, and independent playback controls enhance user flexibility and personalization. These capabilities ensure that users can tailor the viewing experience according to their comprehension preferences.

Advantages of Proposed System

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

The proposed system offers several advantages over existing accessibility solutions. Caption-driven sign language rendering enables automatic generation of sign language interpretation from uploaded captions, reducing manual effort. Parallel video playback allows users to watch both the primary video and sign language interpretation simultaneously, improving comprehension. User playback controls provide flexibility by allowing independent adjustment of speed, pause, and synchronization delay. Additionally, the platform simplifies accessibility integration for content creators by allowing them to upload videos with captions, enabling automated sign language support without requiring specialized production resources. These features collectively enhance accessibility, scalability, and usability for deaf and hard-of-hearing users.

Functional Requirements

The functional requirements define the core operations of the Sign-Language Media Hub system. The platform is designed with three primary modules: User Module, Admin Module, and Content Creator Module, each responsible for specific functionalities to ensure smooth system operation.

The User Module enables users to register and log in to the system securely. After authentication, users can search for videos and configure accessibility preferences according to their requirements. The platform allows users to play videos with synchronized sign language display and provides playback controls such as pause, speed adjustment, and delay customization. These features enhance the viewing experience and improve accessibility. Users can also log out securely after completing their session.

The Content Creator Module enables content providers to register and log in to the platform. After authentication, creators can upload videos and attach captions required for sign language generation. The system allows creators to manage video details, including editing descriptions, updating captions, and modifying metadata. These features simplify the process of creating accessible content while maintaining flexibility and control.

Non-Functional Requirements

Non-functional requirements define the system performance characteristics and operational constraints. Performance is a critical factor, and the system is designed to automatically scale resources based on varying workloads. Monitoring tools continuously track system performance and detect issues, minimizing the need for manual intervention.Security is implemented through automated threat detection and routine security checks. Alerts are generated for critical security events, while administrators intervene only when necessary to address confirmed threats or policy changes. Reliability is achieved through automated failover mechanisms and self-recovery features, ensuring minimal downtime and consistent system availability.Usability is another key requirement, emphasizing intuitive design, easy navigation, and efficient interaction. The platform is designed to provide a user-friendly experience that supports learnability, efficiency, and user satisfaction. Scalability is also incorporated to allow the infrastructure to handle increased user demand and large-scale content growth without performance degradation.

Computational Details

Software Requirements

Software requirements define the technological environment necessary for system development and deployment. These requirements describe what the system should accomplish rather than specifying implementation details. They provide a foundation for cost estimation, development planning, and progress tracking throughout the project lifecycle.The system operates on the Windows operating system and utilizes Node.js as the web server environment. The user interface is developed as a web-based platform to ensure accessibility across devices. The backend and frontend technologies include Next.js 15, React, TypeScript, Tailwind CSS, and Python for artificial intelligence integration. The system requires internet connectivity for API communication and cloud-based processing. SQLite is used as the database to manage user data, video information, and caption metadata.

Hardware Requirements

Hardware requirements specify the physical computing resources required for system operation. The recommended configuration includes an Intel Core i5 processor operating at 1.1 GHz or higher, 8GB of RAM for efficient multitasking, and a minimum of 512GB hard disk storage to support video processing and database management. These specifications ensure smooth performance during development and deployment.

Life Cycle Model

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

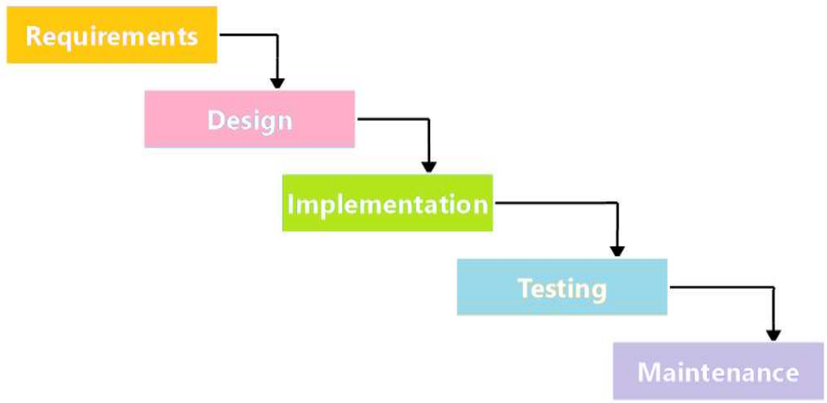

Fig 1: Life cycle model

The Sign-Language Media Hub follows the Waterfall Model, which provides a structured and sequential approach to software development. In the requirement analysis phase, system requirements such as video uploading, caption integration, and sign language interpretation are identified and documented. The design phase defines system architecture, including frontend interfaces, backend processing, and sign language generation modules.During the implementation phase, individual components such as video streaming, caption processing, and sign display modules are developed. These components are integrated in the integration phase, followed by comprehensive testing to verify synchronization accuracy, functionality, and usability. After successful testing, the system is deployed for real-world usage, allowing users to access accessible video content. Maintenance activities are conducted post-deployment to enhance system performance and address future requirements.illustrates the Waterfall Life Cycle Model followed in the development of the Sign-Language Media Hub system.

Design

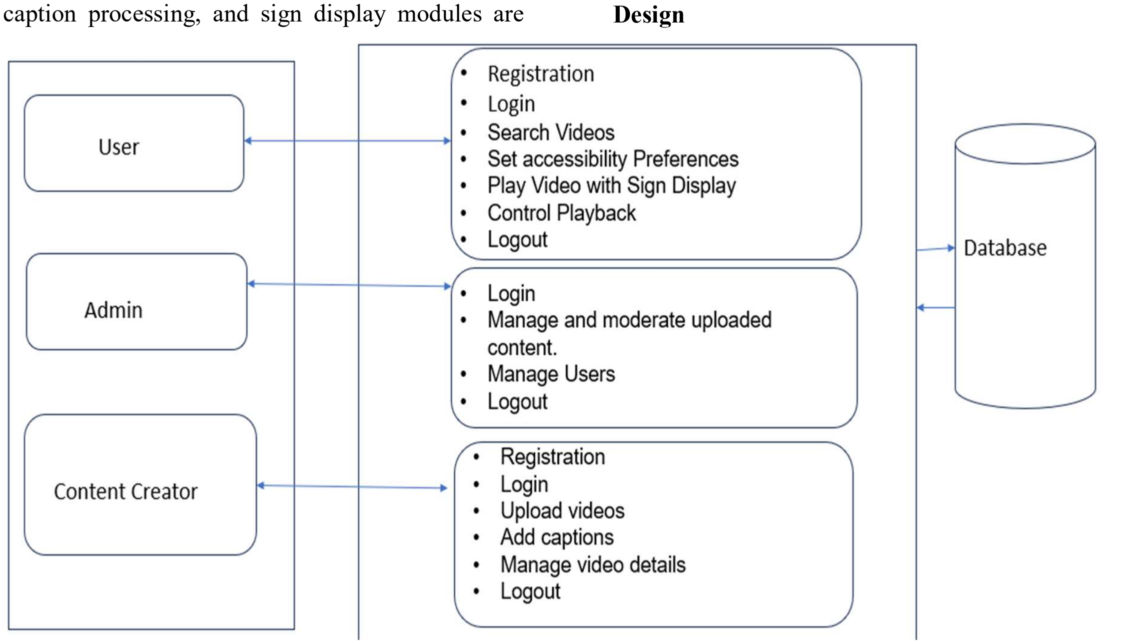

Fig. 2 Software Architecture

Project architecture defines the structural organization of the system, the components involved, and the flow of request processing among these components. It provides a formal representation that supports reasoning about system structure, behavior, and scalability. The architecture for the Sign-Language Media Hub consists of both software architecture and technical architecture. These architectural layers collectively ensure efficient content delivery, accessibility, and scalability while supporting multiple user roles and AI-based processing.

Software Architecture

The software architecture of the Sign-Language Media Hub is designed around three primary roles: User, Content Creator, and Admin. Each role performs specific responsibilities within the platform while sharing core authentication and data access mechanisms. Users interact with the system by registering, logging in, searching for videos, and

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

watching multimedia content enhanced with synchronized sign language representation. The platform provides accessibility features that allow users to customize their viewing preferences, including caption display and sign-language overlays.All system components interact with a centralized database that stores user information, video metadata, captions, gesture mappings, and accessibility preferences. The system follows a multi-tier architecture consisting of a presentation layer, application layer, and data layer. The presentation layer handles user interaction, the application layer processes business logic and AI-based transformations, and the data layer manages persistent storage. This layered architecture improves scalability, maintainability, and system performance.

Technical Architecture

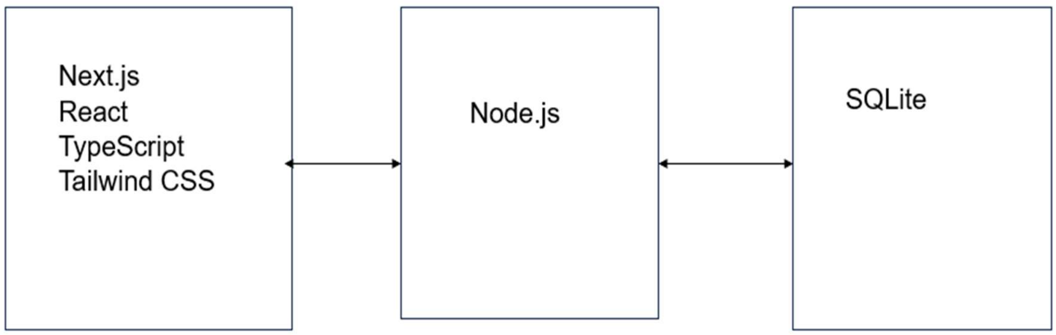

Fig.3 Technical Architecture

The technical architecture of the Sign-Language Media Hub follows a three-tier architecture model. The frontend layer is developed using Next.js with React, TypeScript, and Tailwind CSS. This layer provides responsive user interfaces and accessibility-focused design for smooth user interaction.The backend layer is implemented using Node.js, which handles API communication, business logic, authentication, video processing workflows, and AI model integration. The backend also coordinates communication between the frontend interface and the data storage layer.

Use Case Diagram

The Sign-Language Media Hub includes three main actors: User, Content Creator, and Admin. Each actor interacts with the system based on defined permissions and responsibilities.Users can browse videos, search content, view sign-language enabled media, and save videos for future access. The system allows users to customize accessibility preferences, improving usability for individuals with hearing impairments.Content Creators can upload new videos, tag content, add captions, and manage their own contributions. These features allow creators to continuously expand the platform's accessible content repository.Admins oversee system functionality by managing users, approving uploaded content, moderating inappropriate material, and organizing categories and tags. These operations maintain platform reliability and accessibility quality.

Class Diagram

The class diagram of the Sign-Language Media Hub includes three main classes: Admin, User, and Content Creator. These classes share common attributes such as username, email, and password, which are used for authentication and account management.The Admin class includes functionalities such as login, user management, and content moderation. The User class includes features such as registration, searching videos, setting accessibility preferences, playing videos, controlling playback, and logout functionality. The Content Creator class includes features such as registration, login, video uploading, caption creation, video editing, and logout functionality.This class structure promotes modularity while maintaining shared authentication mechanisms across different roles.

Sequence Diagram

The sequence diagram illustrates how system components interact during a user request. The process begins when a user enters a search query in the frontend interface. The request is transmitted to the backend API, which processes the query and retrieves matching results from the database. The database returns the relevant data to the backend, which forwards the results to the frontend for display.When the user selects a video, the frontend requests playback details from the backend. The backend retrieves video URLs, captions, and gesture mapping data from the database or storage system. The frontend then streams the video to the user and displays synchronized captions and sign-language animations. This interaction demonstrates the communication between user interface, backend processing, and data storage components.

Algorithms

Random Forest Classifier

The Random Forest classifier is used for sign recognition. Initially, training data consisting of feature vectors and labels representing sign gestures is collected. Multiple decision trees are created using bootstrap sampling, where random subsets of data and features are selected. Each tree is trained by splitting data based on feature conditions until stopping criteria are reached. All trained trees are combined to form the Random Forest ensemble.During prediction, a new feature vector representing hand landmarks is passed through each tree. Each tree produces a classification output. The system then performs majority voting, and the class with the highest number of votes is selected as the final prediction.

Media Pipe Hands

MediaPipe Hands is used to extract hand landmarks from video frames. The process begins by capturing video frames and detecting hands using a machine learning model. The system extracts 21 landmark points, each containing x, y, and z coordinates. These coordinates are combined into a feature vector. For example, if three landmarks are considered, their coordinates are combined sequentially into a numerical vector. This feature vector is then passed to the Random Forest classifier for gesture prediction.

OpenAI Whisper

OpenAI Whisper is used for speech-to-text caption generation. Audio is extracted from uploaded videos and converted into waveform format. The waveform is then transformed into a spectrogram representing time-frequency information. A neural network processes the spectrogram and predicts a sequence of words. Each predicted word is assigned timestamps, and the output is formatted into caption data. This caption data is used to synchronize sign-language display with video playback.

Pseudocode and Processing Workflow

The system begins with user registration, where user details are validated and stored in the database. During login, credentials are verified before granting access. When videos are uploaded, captions are generated using speech-to-text processing. The system extracts frames from the video, preprocesses frames, and detects hand landmarks. These landmarks are converted into feature vectors, which are classified using the Random Forest model.The Sign-Language Media Hub architecture ensures scalable performance, accessibility compliance, and efficient content processing. The integration of Random Forest classification, MediaPipe hand tracking, and Whisper speech recognition enables automated sign-language generation, making multimedia content accessible to hearing-impaired users.

Testing

Software testing plays a critical role in validating the functionality, reliability, and performance of the Sign Language Media Hub. Since the system is designed to convert multimedia content into sign-based representations for deaf and mute users, accuracy and synchronization are essential. The testing process ensures that each module operates according to system requirements and delivers meaningful outputs for end users.The media processing module undergoes testing to verify synchronization between video playback, captions, and generated sign representations. User interface testing ensures smooth navigation, accessibility compliance, and ease of interaction for users with hearing impairments. Backend testing validates communication between services, while database testing confirms accurate storage and retrieval of user data, captions, and media files.

Conclusion

The proposed **Sign-Language Media Hub** addresses the growing need for accessibility in digital media platforms by integrating sign language interpretation directly within video content. The system enhances inclusivity by combining synchronized sign language displays, captions, and customizable playback controls. These features enable individuals with hearing impairments to access multimedia content more effectively and independently.Furthermore, the platform benefits content creators by providing tools to embed accessibility features without requiring extensive technical expertise. The integration of sign language alongside traditional subtitles improves communication clarity and promotes equal participation in digital environments. Overall, the Sign-Language Media Hub contributes to building a more inclusive media ecosystem by reducing communication barriers and supporting diverse user needs.

Future Scope

The proposed system offers several opportunities for future enhancements and expansion. One potential improvement is the implementation of real-time sign language conversion for live video streaming and online conversations. This feature would significantly enhance accessibility during live events, educational sessions, and virtual meetings.Additionally, incorporating multi-language support can broaden the platform's reach and make it useful for users across different regions and linguistic backgrounds. Developing a dedicated mobile application would further improve accessibility by enabling users to access content conveniently on smartphones and tablets.In the future, the Sign-Language Media Hub can evolve into a comprehensive accessibility platform that supports inclusive communication across education, entertainment, and professional environments.

9. References

1) H. Bull, T. Afouras, G. Varol, S. Albanie, L. Momeni, and A. Zisserman, “Aligning

ISSN 2347–3657

Volume 14, Issue 1, 2026

subtitles in sign language videos," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, 2024, pp. 11552–11561.

2) K. Papadimitriou and G. Potamianos, "Sign language recognition via deformable 3D convolutions and modulated graph convolutional networks," in *ICASSP*, 2025.
3) N. Sarhan and S. Frintrop, "Transfer learning for videos: From action recognition to sign language recognition," in *Proc. Int. Conf. Image Process. (ICIP)*, 2024.
4) Y. Li, X. Wang, and J. Zhang, "Spatial-temporal enhanced network for continuous sign language recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 3, pp. 456–469, Mar. 2025.
5) M. Ahmed, S. Khan, and R. Ali, "Automatic subtitle synchronization and positioning system dedicated to deaf and hearing-impaired people," *IEEE Access*, vol. 11, pp. 12034–12045, 2023.
6) T. Huang, L. Chen, and P. Singh, "Real-time sign language translation using deep neural networks," in *Proc. ACM Multimedia*, 2024, pp. 332–340.
7) A. Verma and S. Gupta, "Multi-language support in sign language interpretation platforms," *J. Multimodal User Interfaces*, vol. 19, no. 2, pp. 101–115, 2025.
8) R. Das, V. Patel, and H. Mehta, "Mobile applications for accessible media: A review for hearing-impaired users," *Int. J. Human-Computer Interaction*, vol. 42, no. 4, pp. 355–370, 2024.
9) F. Oliveira and M. Silva, "AI-driven sentence-to-sign translation for natural communication," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 5, pp. 2345–2358, 2025.
10) J. Thompson, L. Wang, and K. Roberts, "Enhanced video accessibility through integrated sign language and captions," in *Proc. Int. Conf. Assistive Technol.*, 2023, pp. 89–96.