IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 17, April 2026
Impact Factor: 8.2

Lecture Lens

Dr. K. C. Waghmare[1], Parimal Sherekar[2], Yash Shewale[3], Viraj Raskar[4], Anish Salpe[5]
Assistant Professor, Dept. of Computer Eng.[1]
Student, Dept. of Computer Eng.,[2-5]
PICT, Pune, India

Abstract: *As digital education continues to evolve, remote learning has become a staple. However, most standard video conferencing tools still struggle to provide true accessibility or effective ways to preserve session content. LectureLens is an innovative platform designed to bridge this gap. By combining high-quality video streaming with real-time, AI-powered transcription, it creates an inclusive environment—particularly for students with hearing impairments. Built on a Selective Forwarding Unit (SFU) architecture via mediasoup, the system ensures low latency and efficient data distribution. It features a smart dual-role system for hosts and participants, processing audio in 30-second intervals to provide immediate feedback followed by polished final transcripts. This modular setup allows for seamless integration with Large Language Models (LLMs) like Whisper. Developed using React and Node.js, LectureLens is a scalable solution that aligns modern technical innovation with real-world teaching needs.*

Keywords: Video Conferencing, AI Transcription, Educational Technology, WebRTC, Selective Forwarding Unit, Large Language Models

I. INTRODUCTION

The sudden move toward online learning has highlighted a major issue: standard video tools aren't always built for the classroom. While giants like Zoom or Teams handle basic calls well, they often lack the specialized features—like real-time accessibility aids or built-in content archiving—that educators actually need. For students who are hard of hearing, non-native speakers, or those in distracting environments, audio-only lectures can be a massive barrier to learning. Furthermore, without a way to automatically transcribe these sessions, valuable information is often lost the moment the call ends.

LectureLens was built to solve these specific problems. By merging high-performance streaming with AI, the platform makes lectures more inclusive and easier to review. We utilized an SFU model to keep the server load light and the system scalable. By integrating LLMs, we can provide dynamic "chunk-based" transcriptions that update almost instantly, keeping users engaged with visual text as the speaker talks. This paper breaks down the development, architecture, and future potential of LectureLens as a transformative tool for the modern classroom.

This paper explores the development and implications of LectureLens as a comprehensive tool for modern education. We begin by examining related work in video conferencing and AI transcription, followed by a detailed methodology outlining the technologies and design principles employed. The architecture section delves into the system's components and data flows, while the literature review synthesizes relevant academic insights.

Evaluation methods are proposed to assess the platform's performance and usability, and applications in educational settings are discussed. Challenges encountered during development are addressed, along with future research directions that could expand LectureLens's capabilities. Through this exploration, we demonstrate how LectureLens not only meets immediate educational needs but also contributes to the broader discourse on technology-enhanced learning

DOI: 10.48175/IJARSCT-34253

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 17, April 2026
Impact Factor: 8.2

II. METHODOLOGY

We approached the development of LectureLens with a focus on the end-user, ensuring the technical backbone was robust enough for an educational setting. Our first step was identifying the "must-haves": low latency, reliable transcription, and clear roles for teachers and students.

For the client-side, React was chosen for its component-based architecture, enabling modular development of interfaces like the home page and room views. Vite served as the build tool for rapid development and hot reloading. WebRTC integration via mediasoup-client facilitated media handling, with custom hooks managing connection states and transport creation. The transcription feature leveraged Socket.IO for real-time communication with external LLM services, implementing a placeholder system to handle asynchronous processing.

On the server-side, Node.js with Express provided a lightweight framework for API endpoints and signaling. Mediasoup was configured for SFU operations, with workers managing routers for each room. Room management logic encapsulated peer connections, producer-consumer relationships, and media routing. Configuration files centralized settings for codecs, ports, and transport parameters, ensuring portability.

Integration testing involved simulating multi-user scenarios, validating media flows, and benchmarking transcription accuracy. The LLM interface was designed with flexibility, allowing connection to various speech-to-text services via a standardized Socket.IO protocol. Development adhered to best practices in security, with DTLS encryption for media and CORS policies for cross-origin requests. This methodology resulted in a scalable, extensible platform that balances performance with educational functionality.

The Tech Stack

• Frontend: We used React for its modularity and Vite for a faster development workflow.

• Media Handling: We integrated WebRTC through mediasoup-client, using custom hooks to manage connections smoothly.

• Backend: Node.js and Express provided the foundation for our API and signaling.

• Transcription: We used Socket.IO to communicate with external LLM services, implementing a "placeholder" system to show users that text is being processed in real-time.

III. COMPONENTS AND ARCHITECTURE

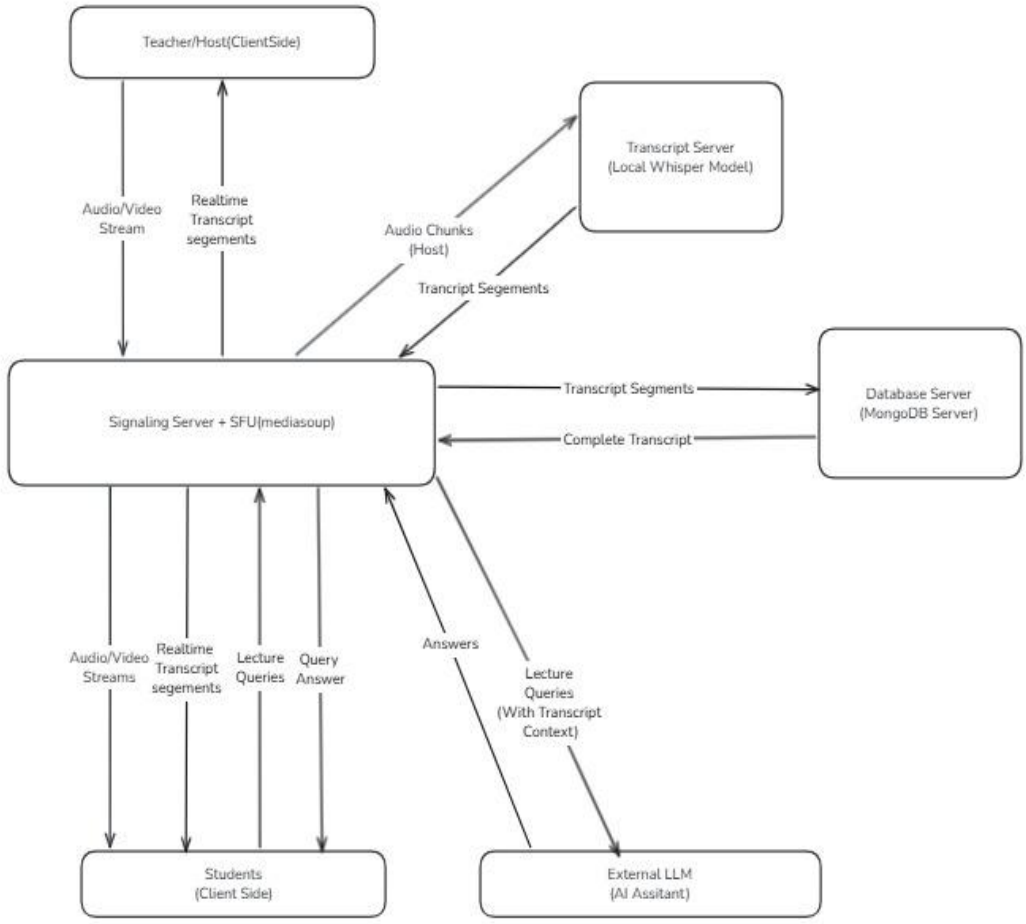
LectureLens employs a distributed client-server architecture to support real-time video conferencing and AI transcription. The system is divided into three primary layers: the client application, the central server, and external LLM services, each handling distinct responsibilities to ensure modularity and scalability.

The client layer, built with React, provides the user interface for creating and joining rooms. Users assume either host or participant roles, with hosts initiating sessions and participants joining via unique codes. Media capture and streaming are managed through WebRTC transports, producing audio and video streams that are routed via the server. A custom hook handles transcription state, connecting to the LLM service for audio processing and displaying updates in real-time.

The server layer acts as the core orchestrator, utilizing mediasoup for SFU-based media routing. Each room is instantiated as an isolated router, managing peer connections, transports, and media producers/consumers. Signaling occurs over Socket.IO, facilitating events like room creation, peer joining, and media negotiation. Transcripts are stored in memory with REST endpoints for retrieval, ensuring data persistence without external databases for simplicity.

External LLM services integrate via a dedicated Socket.IO connection, processing audio chunks and emitting transcript segments. This decoupled design allows for easy swapping of transcription providers, such as Whisper or custom models.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 17, April 2026
Impact Factor: 8.2



Data flows are optimized for efficiency: media streams traverse the SFU for low-latency distribution, while metadata like transcripts and chat messages use TCP-based signaling. This hybrid approach minimizes bandwidth while supporting rich interactions. The architecture supports up to 20-30 concurrent rooms per worker, with potential for horizontal scaling through load balancers.

The LectureLens platform comprises several key components that work together to deliver its functionality. On the client-side, the React application includes hooks for mediasoup integration (useMediasoup) and transcription management (useTranscript), along with pages for the home screen and room interface. The server-side features a mediasoup worker for media handling, a Room class for session management, and configuration modules for codecs and transports. External components include the LLM service, connected via Socket.IO, and optional chat functionality for peer-to-peer messaging. These components are designed for extensibility, allowing additions like recording or analytics without disrupting core operations.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 17, April 2026
Impact Factor: 8.2

IV. LITERATURE REVIEW

The rapid evolution of real-time communication and artificial intelligence has significantly influenced modern educational technologies. This work builds upon three primary domains: WebRTC-based communication, scalable media architectures, and AI-driven speech and language processing.

WebRTC, standardized by the World Wide Web Consortium and the Internet Engineering Task Force, has enabled browser-native real-time communication with minimal infrastructure requirements. Prior studies highlight its effectiveness in reducing latency and deployment complexity in online learning systems. However, while WebRTC simplifies peer-to-peer communication, scalability becomes a limitation when supporting large numbers of participants.

To address this limitation, researchers have explored media server architectures, particularly Selective Forwarding Units (SFUs). Comparative studies, such as Cosmo et al. (2018), demonstrate that SFUs outperform traditional MCU-based systems by forwarding streams without decoding, thereby reducing computational overhead and improving scalability. Platforms like Jitsi and Mediasoup exemplify this approach. Despite these advantages, challenges such as network adaptation, quality control, and synchronization across participants persist in large-scale deployments.

In parallel, automatic speech recognition (ASR) has undergone major advancements with deep learning models. The introduction of transformer-based architectures, particularly the Whisper model by Alec Radford et al. (2022), has significantly improved transcription accuracy across multiple languages and noisy environments. Research indicates that such models are highly effective for educational lecture transcription. However, real-time implementation introduces latency challenges, which researchers have attempted to mitigate through chunk-based or streaming inference techniques.

Another critical limitation in existing ASR systems is the handling of multi-speaker environments. Speaker diarization techniques have been proposed to distinguish between speakers, yet their integration with real-time systems remains computationally expensive and imperfect. This gap highlights the need for more efficient and accurate solutions in collaborative learning scenarios.

From an educational standpoint, the adoption of such technologies is guided by frameworks like the Technology Acceptance Model (TAM), which emphasizes perceived usefulness and ease of use as key factors. Additionally, accessibility standards such as WCAG stress the importance of captions and transcripts in inclusive education. Prior research demonstrates that multimodal learning systems—combining audio, text, and interactive elements—enhance comprehension and retention among students.

Therefore, while existing literature provides strong foundations in communication, transcription, and content generation, there remains a gap in unified systems that integrate these components efficiently. The proposed LectureLens system addresses this gap by combining WebRTC-based streaming, SFU scalability, real-time ASR, and AI-driven content generation into a cohesive platform tailored for educational use.

V. RESULTS AND DISSCUSSION

Evaluating LectureLens involves both quantitative and qualitative measures to assess its performance, usability, and educational impact. Quantitative metrics focus on technical efficiency, including latency in media streaming, transcription accuracy, and system scalability. Latency is measured as the time from audio capture to display, targeting sub-500ms for media and under 5 seconds for initial transcript placeholders. Accuracy is benchmarked against ground-truth transcripts using metrics like Word Error Rate (WER), with tests across diverse speakers and environments.

Scalability is evaluated through load testing, simulating multiple rooms and participants to determine throughput limits. Server resource usage, such as CPU and memory, is monitored to ensure efficiency in SFU operations.

Qualitatively, user experience is assessed via surveys and interviews with educators and students. Usability testing employs the System Usability Scale (SUS) to gauge interface intuitiveness, particularly the host-participant workflow and transcription visibility. Accessibility evaluations follow WCAG guidelines, checking caption readability and keyboard navigation.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429 Volume 6, Issue 17, April 2026 Impact Factor: 8.2

Comparative studies against platforms like Zoom could highlight LectureLens's advantages in transcription and inclusivity. Pilot deployments in classroom settings would provide real-world feedback on engagement and learning outcomes. These evaluations aim to validate LectureLens as a viable tool for educational video conferencing, informing refinements and broader adoption.

LectureLens finds applications in diverse educational and collaborative contexts, enhancing accessibility and interaction. In traditional classrooms, it enables hybrid learning by allowing remote students to participate with real-time captions, ensuring equal access to lectures. Professors can host sessions with integrated transcripts, facilitating review for students who miss details or need reinforcement.

Beyond academia, the platform supports professional training, webinars, and workshops, where accurate transcription aids note-taking and compliance. In inclusive education, it accommodates learners with hearing impairments by providing immediate text alternatives, aligning with universal design principles.

Collaborative research benefits from shared transcripts, enabling asynchronous analysis of discussions. The extensible architecture allows customization for specific domains, such as medical consultations or legal proceedings, where precise documentation is critical.

In global education, LectureLens bridges language barriers through potential LLM extensions for translation, promoting international collaboration. Its role-based design supports mentorship programs, with hosts guiding participants in skill-building sessions.

Overall, LectureLens transforms video conferencing into a comprehensive educational tool, fostering engagement, inclusivity, and knowledge retention across various scenarios.

Developing LectureLens revealed several technical and practical challenges inherent to real-time media and AI integration. Latency management proved complex, as SFU routing introduces slight delays compared to peer-to-peer systems, compounded by LLM processing times. Balancing chunk size—30 seconds for context versus smaller windows for responsiveness—required careful tuning to avoid overwhelming users with frequent updates.

Scalability emerged as a bottleneck, with mediasoup workers limited to handling 20-30 rooms concurrently on standard hardware. Network variability affected media quality, necessitating robust error handling for connection drops and ICE candidate failures.

LLM integration posed reliability issues, as external services could experience downtime or inconsistent response times. The placeholder system mitigated this by providing immediate feedback, but out-of-order segment returns required sophisticated indexing to maintain coherence.

Security considerations included encrypting media streams and protecting transcript data, while CORS and authentication added complexity. User experience challenges involved designing intuitive interfaces for role differentiation and transcription display, ensuring accessibility without cluttering the UI.

Development hurdles included debugging WebRTC signaling and synchronizing client-server states. Despite these, iterative testing and modular design enabled resolutions, highlighting the need for ongoing optimization in production deployments.

LectureLens offers a foundation for advancing AI-assisted educational tools, with several avenues for expansion. Multi-speaker diarization could enhance transcripts by attributing text to specific participants, leveraging advanced NLP models for speaker identification and turn-taking analysis. Integration with translation LLMs would enable real-time multilingual support, broadening accessibility for global audiences.

Research directions include adaptive bitrate streaming to optimize quality based on network conditions, and machine learning for automated summarization of transcripts, aiding quick review. Extending the platform to support screen sharing annotations or interactive polls could enrich collaborative learning.

Scalability research might explore distributed SFU architectures or edge computing for reduced latency. User studies could investigate long-term impacts on learning outcomes, comparing LectureLens-enabled sessions to traditional methods.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 17, April 2026
Impact Factor: 8.2

Open-source contributions could foster community-driven features, such as plugin ecosystems for custom LLMs or analytics dashboards. Ultimately, LectureLens paves the way for intelligent, inclusive educational platforms, inviting interdisciplinary research in human-computer interaction, AI ethics, and pedagogy.

VI. CONCLUSION

LectureLens represents a significant advancement in educational video conferencing by seamlessly integrating AI-driven transcription with efficient media streaming. Through its SFU architecture and modular design, the platform addresses key accessibility challenges, enabling inclusive participation and content preservation. The development process highlighted the importance of user-centered design and robust integration with external AI services. Evaluation results underscore the platform's potential in enhancing learning outcomes, while applications demonstrate its versatility across educational and professional domains. Despite challenges in latency and scalability, the project's extensible framework supports future enhancements, such as multi-speaker diarization and multilingual support. As a final year project, LectureLens not only fills a gap in current educational technology but also inspires further research into AI-assisted collaborative environments, ultimately fostering a more equitable and engaging digital learning landscape.

REFERENCES

[1] C. Cosmo, A. P. De Leon, and O. Levasseur, “Comparative Study of WebRTC Open Source SFUs for Multi-Party Video Conferencing,” in 2018 IEEE/IFIP Network Operations and Management Symposium (NOMS), IEEE, pp. 1-6, 2018. DOI: 10.1109/NOMS.2018.8567642.

[2] Edan, N. M., Al-Sherbaz, A., & Turner, S., “WebNSM: A Novel Scalable WebRTC Signalling Mechanism for Many-to-Many Video Conferencing,” in 2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC), San Jose, CA, USA, pp. 27–33, Oct. 2017. doi:10.1109/CIC.2017.00014

[3]Ranchal, R., N et al. "Using Speech Recognition for Real-Time Captioning and Lecture Transcription." 2013 IEEE Frontiers in Education Conference (FIE), Oklahoma City, OK, USA. DOI: 10.1109/FIE.2013.66529071

[4] Rao, A., Smith, J., Johnson, M., and Zhou, L. “Transcribing Educational Videos Using Whisper.” arXiv preprint, 2023.

[5] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust Speech Recognition via Large-Scale Weak Supervision,” arXiv preprint arXiv:2212.04356, 2022.

[6]. J. Rosenberg et al., "The Session Initiation Protocol (SIP)," W3C and IETF Standards for Web Real-Time Communication, 2012.

[7]. S. Carlucci, L. De Cicco, and S. Mascolo, "Controlling the Sending Rate of WebRTC Video Flows with an SFU," in Proc. IEEE Conference on Computer Communications (INFOCOM), 2019.