WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

An AI-Enabled Smart Blackboard for Real-Time Lecture Transcription and Automated Study Notes

Harini B, Guna Soundarya M, Hariharasudan V S, Hari Dharsan J.*

BE.CSE, Sri Shakthi Institute of Engineering and Technology, Coimbatore and 641062, India.
*E-mail address: harinibabu113@gmail.com

A B S T R A C T

Classroom learners often need to divide their attention between understanding a lecture and recording its key points. This paper presents AI Smart Board, a web-based prototype intended to support lecture capture and revision by combining a digital blackboard interface, speech-to-text input, AI-assisted summarization, and saved study notes. The proposed workflow allows a teacher or learner to capture spoken content as editable text, request a concise studyoriented summary, and retain the resulting notes for later review. The current implementation uses a React/Vite user interface, an Express-based service layer for AI requests, and browser-side storage for saved notes. The system is designed as a lightweight prototype rather than a replacement for teacher-authored material. Because the proposed evaluation has not yet been carried out, no numerical accuracy, latency, or learning-gain claims are made. The paper defines an evaluation plan using word error rate for transcription, human-rated factual consistency and coverage for summaries, response-time measurements, and task-based usability testing. This work provides a reproducible system description and a basis for empirical evaluation of integrated lecture-note support.

Keywords: Educational technology; intelligent learning support; lecture transcription; speech recognition; text summarization; web application

1. Introduction

Taking notes during a lecture requires learners to listen, interpret information, identify important concepts, and write at the same time. When attention is directed toward transcription, some explanations or examples may be missed. Speech recognition and automatic summarization can help make spoken material easier to revisit, although transcription errors and generated-summary omissions remain important limitations (Blunt & Haskins, 2021, 2025).

This paper describes AI Smart Board, a browser-based prototype that combines a board-style writing interface with speech-to-text capture and AIassisted study-note generation. Unlike a standalone transcription utility, the prototype places captured text in an editable board workflow and provides a path to save the processed notes for later revision. The system is intended to assist teachers and students; users remain responsible for checking the transcript and generated summary against the original lesson.

The contributions of this paper are: (1) a description of an integrated web workflow for capturing lecture text, editing it, summarizing it, and saving study notes; (2) a practical architecture using a React/Vite client and an Express service for AI requests; and (3) a proposed evaluation protocol that avoids unsupported claims and measures transcription, summary quality, response time, and usability. The prototype itself is not claimed to establish improved learning outcomes; that question requires a separate study. The remainder of the paper is organized as follows. Section 2 reviews related work, Section 3 presents the system overview, Section 4 describes the implementation, Section 5 proposes the evaluation framework, Sections 6 and 7 discuss constraints and future work, and Section 8 concludes.

2. Related work

Prior work has investigated automatic speech recognition as a way to generate lesson transcripts and summaries. Blunt and Haskins (2021) describe a model for applying automatic speech recognition to lesson-summary generation and discuss the effect of classroom noise on the pipeline. More recent work has also examined speech-driven lesson summary generation, comparing recognition systems and highlighting the importance of testing under noisy conditions (Blunt & Haskins, 2025).

Lecture summarization has also been explored using neural sequence models. Research on attention-based recurrent neural networks has investigated segmentation and content selection for lecture-video transcript summarization (Automatic Lecture Video Content Summarization, 2019). These studies show that transcription and summarization are established research directions; therefore, integrating similar functions into a web application alone should not be presented as a novel algorithm.

AI Smart Board focuses on the application workflow: a board-like editing environment, transformation of captured text into study notes, and persistence of notes for later use. A fair evaluation should compare its output with the source transcript or instructor-prepared notes and document the limits of the current prototype. Further work is required to determine whether the integrated interface offers measurable advantages over separate transcription and summarization tools. Table 1 contrasts the cited studies with the present work.

Table 1 - Comparison of related work and AI Smart Board.

Work	Input	Main focus
Blunt & Haskins (2021)	Classroom speech	ASR model for lesson summaries; noise management
Blunt & Haskins (2025)	Classroom speech	Speech-driven lesson summaries; comparison of recognition systems
Automatic lecture video content summarization (2019)	Lecture video transcripts	Segmentation and content selection with attention-based RNN
This work	Lecture speech and editable text	Board workflow, AI-assisted study notes, saved notes

3. System overview

3.1 Architecture and workflow

The prototype is organized into a client interface, a server-side AI-request layer, and browser-side note persistence. The client is implemented with React and Vite. The Express service receives requests for AI-assisted text processing and communicates with the configured generative-AI service. Saved notes are stored in browser local storage in the current prototype; this supports local persistence on the same browser but does not provide shared multi-user database storage or centralized account management. The overall architecture is shown in Fig. 1.

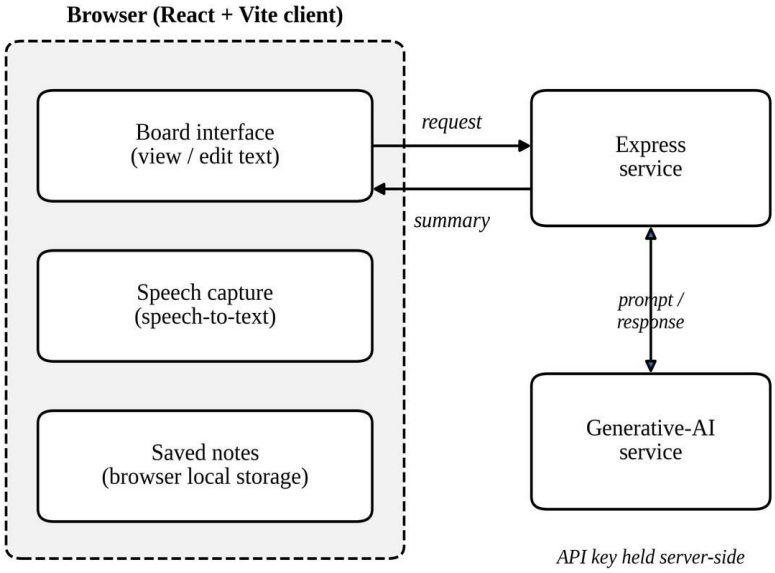


Fig. 1 - Architecture of the AI Smart Board prototype.

The principal workflow, shown in Fig. 2, is: (1) open the board; (2) capture speech as text using the speech-recognition mechanism configured in the deployed browser; (3) inspect and edit the transcript; (4) request an AI-generated summary; (5) review and correct the summary; and (6) save the final study note for later access or export.

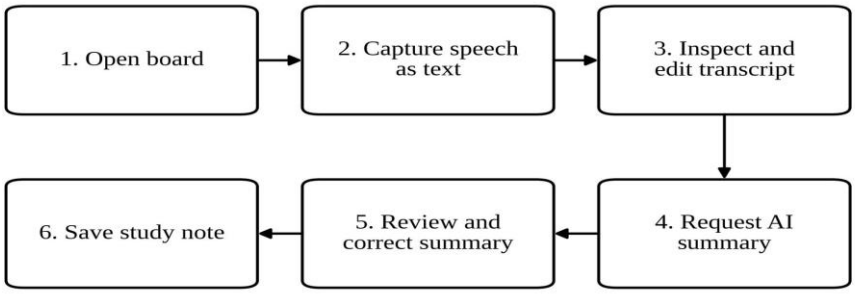


Fig. 2 - Principal workflow of AI Smart Board, from lecture capture to saved study note.

3.2 Functional modules

The prototype is divided into five functional modules, summarized in Table 2. The summarization module is the only one that depends on the server-side AI service; the remaining modules operate in the browser.

Table 2 - Functional modules of AI Smart Board.

Module	Function	Data handled
Board interface	Blackboard-style workspace for viewing and editing lesson text	Transcript text
Speech capture	Converts spoken input to text through the speech-recognition component configured by the application	Speech, text
Summarization	Sends eligible text to the server-side AI service and returns a condensed study-oriented version	Transcript, summary
Saved notes	Retains notes locally for later access	Notes in local storage
Export	Preserves notes outside the active board	Saved notes

3.3 Data and privacy considerations

The current prototype does not provide a centralized database or production-grade authentication. Browser local storage is tied to the local browser profile and can be cleared by the user; it is not a substitute for secure server-side storage. Before classroom deployment, the system should explain whether audio or transcripts are sent to third-party services, minimize retained data, protect API credentials on the server, and obtain appropriate consent for recording.

4. Implementation

The front end uses React with Vite for development and build tooling. An Express-based backend provides an API boundary for AI-assisted summarization, with configuration supplied through environment variables. The generative-AI service is called from the backend rather than exposing its secret key in browser code. Notes are currently retained using browser local storage. Table 3 lists the principal components.

Table 3 - Principal implementation components.

Layer	Technology	Role
Client	React, Vite	Board interface, speech capture, editing, note review
Server	Express	API boundary for summarization; configuration via environment variables
AI service	Configured generative-AI service	Produces condensed study-oriented summaries
Persistence	Browser local storage	Local saved notes on the same browser

The summarization stage is intended to remove low-information conversational filler and reorganize the remaining material into concise study notes. Generated content may omit context or introduce incorrect statements, so the interface allows users to inspect and edit the output before treating it as study material.

5. Evaluation methodology

This section defines a reproducible protocol for evaluating the prototype. The protocol has not yet been executed, and no results are reported in this paper. Fig. 3 gives an overview of the four evaluation dimensions, and Table 4 summarizes the measures and the items to be reported for each.

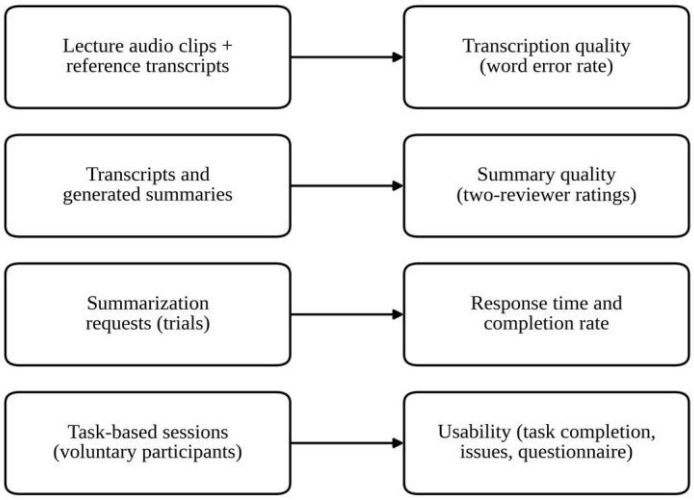


Fig. 3 - Overview of the proposed evaluation framework.

Table 4 - Proposed evaluation dimensions.

Dimension	Measure	Items to report
Transcription	Word error rate, Eq. (1)	Clip count and duration; language, accent, noise conditions
Summary quality	Two-reviewer ratings of consistency, coverage, readability, usefulness	Rating scale; disagreement procedure; example successes and errors
Response time	End-to-end time per request; completion rate	Trials; mean, median, range or SD; network and model configuration
Usability	Task completion, observed issues, questionnaire	Participant count; tasks; approvals and consent

5.1 Transcription quality

Collect a set of lecture audio clips with permission and prepare manually verified reference transcripts. Report the number and duration of clips, language and accent characteristics, recording conditions, and noise conditions. Calculate word error rate (WER) as

$$\mathrm{WER} = (S + D + I) / N \qquad (1)$$

where S, D, and I are substitution, deletion, and insertion counts and N is the number of words in the reference transcript. Report aggregate WER and, where useful, results by condition.

5.2 Summary quality

For each transcript, generate a summary and have at least two reviewers independently rate factual consistency, coverage of key points, readability, and usefulness on a predefined scale. Resolve disagreements using a documented procedure. Include representative examples of both successful summaries and errors. Automatic metrics may supplement but should not replace human review.

5.3 Response time and reliability

Measure end-to-end response time from the user's summarization request until the summary appears. Report the number of trials, average and median response time, range or standard deviation, network conditions, and model configuration. Record successful and failed requests to estimate completion rate. Do not report a single timing result as representative of all conditions.

5.4 Usability

If feasible, conduct a small task-based usability evaluation with voluntary participants and appropriate institutional or instructor approval. Ask participants to complete predefined tasks such as capturing text, editing a transcript, generating a summary, and retrieving a saved note. Report participant count, task completion, observed issues, and questionnaire results. Do not claim that the system improves academic performance without a suitably designed learning study.

6. Discussion

This paper reports a system description and an evaluation framework rather than empirical results. The framework in Section 5 is intended to be applied to the deployed application in follow-up work; until then, the conclusions of this paper are limited to implementation feasibility and design rationale.

The design offers a single workflow for text capture, editing, AI-assisted condensation, and note retrieval. However, several constraints affect its practical value: speech recognition may be sensitive to accents and background noise; generated summaries can omit details or contain inaccuracies; browser local storage does not synchronize notes across devices; and the current architecture should not be assumed to provide robust multi-user authentication or centralized storage. These constraints should be treated as system limitations rather than concealed. Table 5 summarizes these constraints and the corresponding mitigations.

Table 5 - Constraints and mitigations.

Constraint	Mitigation
Recognition sensitive to accents and background noise	Transcript can be inspected and edited before summarization
Summaries may omit details or contain inaccuracies	Summary is reviewed and corrected before saving
Local storage does not synchronize across devices	Planned server-side persistence and note synchronization
No robust multi-user authentication or central storage	Planned role-based teacher and student accounts
Audio and transcripts may reach third-party services	Disclose data flows, minimize retention, obtain recording consent

7. Limitations and future work

The current prototype is an application-level integration and does not introduce a new speech-recognition or summarization algorithm. Its performance and educational value remain unverified until empirical testing is completed. Future work may add secure server-side persistence, role-based teacher and student accounts, note synchronization, multilingual processing, quiz generation, and stronger evaluation using diverse lecture recordings. Any user study should follow relevant institutional requirements and obtain informed consent where required.

8. Conclusion

This paper described AI Smart Board, a web-based prototype that supports lecture capture and revision through a digital board interface, speech-to-text input, AI-assisted summarization, and saved notes. It documented the architecture and workflow and proposed a measurable evaluation framework covering transcription accuracy, summary quality, response time, and usability. Because the evaluation has not yet been carried out, the paper makes no claims about quantified accuracy or learning improvements. Future work will apply the framework to the deployed system and compare it with existing transcription and summarization tools.

REFERENCES

Automatic lecture video content summarization with attention-based recurrent neural network. (2019). In *Proceedings of the International Conference on Artificial Intelligence and Information Technology (ICAIIT)*. https://doi.org/10.1109/ICAIIT.2019.8834514

Blunt, P., & Haskins, B. (2021). A model for the application of automatic speech recognition for generating lesson summaries. *Advances in Science, Technology and Engineering Systems Journal, 6*(2), 526–540. https://doi.org/10.25046/aj060260

Blunt, P., & Haskins, B. (2025). Towards speech-driven lesson summary generation. *Procedia Computer Science, 263*, 884–891.

https://doi.org/10.1016/j.procs.2025.07.106