

Awwazai: A Multilingual AI-Powered Meeting Platform for Indian Languages

Aman Gupta¹, Shreyash Bansode², Anjali Ghangurde³, Keshav Elbhar⁴,
Kamlesh Bhosale⁵, Prof. Deepa Padwal⁶, Prof. Pallavi Chavan⁷

Indira College of Engineering and Management, Pune, India

Abstract: *The usage of meeting transcription and summarization services has seen considerable success internationally, but current commercial offerings still lack behind in catering to the linguistic diversity of India, where interactions often include Hindi, Marathi, English, and even their combinations like Hinglish. In this research, we introduce Awaaz Ai, an open source, AI-driven website capable of transcribing, summarizing, translating, and answering questions based on recorded meetings in multilingual environments in India. Awwazai uses AssemblyAI for detecting languages in audio files, Groq-hosted Llama 3.1 for generating summaries in multiple languages, and FastAPI along with PostgreSQL for storage. An extensive analysis of features in comparison with four commercially available platforms—Otter.ai, Fireflies.ai, Google Meet Captions, and Zoom AI Companion—reveals the uniqueness of Awwaz ai in supporting mixed language transcription, summary creation in detected languages, speaker recognition with personalized tags, and text search over all records. The application is hosted on Render with Neon PostgreSQL database, providing an operational solution for free user access..*

Keywords: Meeting transcription, multilingual NLP, Marathi, Hindi, Hinglish, speech-to-text, AI summarization, speaker diarization, FastAPI, Llama 3.1, AssemblyAI

I. INTRODUCTION

The proliferation of remote and hybrid work models has created a strong demand for automated meeting transcription and summarization tools. While platforms such as Otter.ai, Fireflies.ai, Zoom AI Companion, and Google Meet Captions have achieved widespread adoption in English-speaking markets, their support for the Indian subcontinent's rich linguistic ecosystem remains superficial. India has 22 officially recognized languages and over 780 dialects, with a large proportion of professional and academic discourse occurring in code-switched languages such as Hinglish (Hindi-English) and Marathi-English, where speakers fluidly alternate between languages within a single conversation. Existing tools either transcribe such conversations with significant errors or restrict summarization to English regardless of the source language. This creates a critical usability gap for Indian students, researchers, and professionals who require transcripts and summaries in their native language.

This paper presents Awwazai, a web-based platform that addresses this gap through a layered architecture: AssemblyAI provides audio-level language identification that is agnostic to romanization artifacts in the transcript; Groq's Llama 3.1-8b-instant model generates summaries strictly in the detected dominant language; and a real-time meeting Q&A chat allows users to interrogate past meetings in the language of their choice. Awwazai is fully open-source, deployed on a cloud PaaS, and offered free of charge.

The remainder of this paper is structured as follows: Section II reviews related work and competing platforms; Section III identifies key limitations of existing solutions; Section IV presents the proposed system architecture; Section V describes the implementation; Section VI presents a comparative evaluation; and Section VII concludes the paper. II.



II. RELATED WORK

A. Commercial Transcription Platforms

Otter.ai [1] provides real-time transcription with speaker identification and automated summary generation. Its support is primarily optimized for English, and Indian language transcription quality is inconsistent. Fireflies.ai [2] extends functionality with meeting Q&A (AskFred) but confines intelligent queries to English language responses. Neither platform offers summaries that match the detected language of the input audio.

Google Meet's live captions [3] leverage Google's speech recognition infrastructure and support a broader set of languages including Hindi and Tamil; however, transcripts are ephemeral and no summarization or export functionality is provided. Zoom's AI Companion [4] offers meeting summaries and in-meeting Q&A but is restricted to English and requires an active Zoom subscription.

B. Academic Research on Multilingual ASR

Significant academic effort has been devoted to multilingual automatic speech recognition (ASR). Pratap et al. [5] demonstrated that large-scale self-supervised models such as wav2vec 2.0 generalize to low-resource languages when pre-trained on diverse multilingual corpora. Salesky et al. [6] explored code-switching in speech translation, highlighting the difficulty of processing utterances that alternate between two languages mid-sentence. Khare et al. [7] specifically studied Hindi-English code-switching and noted that romanized transliterations in transcripts defeat script-based language classifiers, a problem directly addressed by Awwazai's audio-level detection strategy.

C. Large Language Models for Summarization

The introduction of instruction-tuned large language models (LLMs) such as GPT-4 [8] and Llama 3 [9] has shifted summarization from extractive to abstractive approaches. Meta's Llama 3.1, accessible via the Groq inference API at sub-second latency [10], enables real-time multilingual summarization without the per-token cost of proprietary APIs. Awwazai exploits this by routing summarization prompts with explicit language instructions derived from the detected audio language, producing outputs in Hindi (Devanagari), Marathi (Devanagari), or English as appropriate.

III. PROBLEM STATEMENT AND LIMITATIONS OF EXISTING SOLUTIONS

A systematic analysis of the four leading commercial transcription tools reveals five critical deficiencies when applied to Indian multilingual contexts:

- Language detection based on transcript text rather than audio phonemes causes consistent misclassification of Marathi as Hindi because both share the Devanagari script and are frequently romanized identically in ASR output.
- Summary generation defaults to English irrespective of the meeting language, rendering the output inaccessible to participants who are not proficient in English.
- Meeting Q&A functionality, where available, is restricted to English queries and responses, excluding speakers of other languages.
- All major platforms operate under subscription-based pricing models, imposing financial barriers on students, small teams, and non-profit organizations.
- None of the reviewed platforms supports post-hoc speaker labeling with persistent custom names or cross-meeting full-text search across transcripts, summaries, and action items simultaneously.

IV. PROPOSED SYSTEM: AWWAZ AI

A. System Overview

Awwazai is a web application comprising a single-page HTML/JavaScript frontend, a Python FastAPI backend, an AssemblyAI integration for transcription, a Groq/Llama 3.1 integration for language-aware summarization and Q&A, and a PostgreSQL database hosted on Neon. The system is accessible at awwazai.onrender.com and supports both file upload (asynchronous background processing) and live microphone recording via WebSocket streaming.



B. Language Detection Architecture

The core innovation of Awwazai is a three-layer language detection strategy designed to overcome the limitations of text-based classifiers for code-switched Indian language transcripts:

- Layer 1 — AssemblyAI audio-level detection: When a file is uploaded, the TranscriptionConfig is invoked with language_detection=True, directing AssemblyAI to infer the dominant language from audio phonemes rather than transcript text. This correctly identifies Marathi from audio even when the romanized transcript text contains no Devanagari markers.
- Layer 2 — English co-detection: After the primary language code is returned, the transcript text is scanned for the proportion of ASCII alphabetic words. If more than 15% of words are exclusively ASCII, English is appended as a secondary language, forming mixed-language pairs such as Marathi+English.
- Layer 3 — LLM fallback: For live WebSocket sessions where no audio-level code is available, a Groq/Llama 3.1 call with an explicit system prompt identifying Marathi versus Hindi linguistic markers serves as a fallback detector. A mixed-language profile dictionary maps detected language sets (e.g., {mr, en}) to human-readable labels (Marathi + English) and language-specific summarization instructions passed directly into the LLM prompt.

C. Multilingual Summarization

Summarization is performed in two stages. For short transcripts (fewer than 3,000 characters), the full text is submitted directly to Groq. For longer transcripts, up to three chunks are processed in parallel using asyncio.gather, reducing wall-clock latency by approximately 3x compared to sequential calls. The final summary call includes an explicit language instruction derived from the detected language profile. Summaries in Hindi and Marathi are generated in Devanagari script; Hinglish summaries are generated in Hindi with English technical terms preserved naturally.

D. Non-blocking Upload Pipeline

To address the latency inherent in cloud transcription services, the /upload endpoint returns a meeting_id immediately with status PROCESSING. AssemblyAI transcription, language detection, and summarization proceed in a FastAPI BackgroundTask. The frontend polls the /meetings endpoint every five seconds and triggers a speaker labeling popup and dashboard refresh upon completion. This architecture eliminates perceived wait time and enables the system to handle multiple concurrent uploads without blocking.

E. Meeting Q&A

Each meeting's full transcript, summary, and speaker label map are loaded as context into a system prompt provided to Groq/Llama 3.1. Speaker codes (e.g., A, B) are replaced with real names before the context is submitted, allowing users to query by speaker name. Up to ten turns of conversation history are appended to each API call, enabling coherent multi-turn dialogue. The interface responds in the language of the user's query.

V. IMPLEMENTATION

The following table summarizes the technology stack employed in Awwazai:

Component	Technology	Purpose
Frontend	HTML5, Tailwind CSS, JS	Responsive single-page dashboard
Backend API	FastAPI (Python)	REST endpoints, WebSocket, background tasks
Transcription	AssemblyAI	Audio-to-text with language detection
Summarization & Q&A	Groq / Llama 3.1-8b	Multilingual summary, action items, chat
Translation	Google Translate API	Free client-side translation (5 languages)
Database	PostgreSQL on Neon	Meetings, users, sessions, speaker labels
Authentication	bcrypt + token sessions	Secure login, registration, session management
Deployment	Render (Cloud PaaS)	Hosted at awwazai.onrender.com

TABLE I. Technology Stack of Awwazai



The frontend is a single HTML file served directly by FastAPI, eliminating the need for a separate frontend build pipeline. Authentication uses bcrypt password hashing with secure random token generation (32-byte URL-safe tokens) stored in a PostgreSQL sessions table with configurable expiration. All database connections include SSL enforcement for compatibility with Neon's serverless PostgreSQL. The system has been deployed on Render's free tier with Neon's free PostgreSQL instance, demonstrating production viability at zero infrastructure cost.

The Awaaz AI platform is built using a modern full-stack architecture that ensures scalability, efficiency, and real-time performance. The frontend is developed using HTML and Tailwind CSS, providing a responsive and user-friendly dashboard for managing transcripts, summaries, and meeting data. The backend is powered by FastAPI, a high-performance Python web framework that enables asynchronous processing and seamless API integration. FastAPI handles core functionalities such as audio upload, real-time transcription streaming, authentication, and data management, ensuring low latency and high throughput.

The system integrates multiple external APIs and libraries to enhance its capabilities. Speech-to-text conversion is performed using AssemblyAI, which provides accurate real-time transcription. For summarization and natural language understanding, the system utilizes OpenAI (via Groq API), enabling intelligent summary generation and action item extraction. Additionally, Google Translate API is used for multilingual translation, supporting languages such as English, Hindi, and Marathi. Data is stored using SQLite/PostgreSQL, while ReportLab is used for generating PDF summaries. This combination of technologies allows Awaaz AI to deliver a robust, scalable, and feature-rich solution for real-time meeting intelligence.

VI. COMPARATIVE EVALUATION

Table II presents a feature-by-feature comparison of Awwazai against the four leading commercial transcription and meeting intelligence platforms. Evaluation criteria were selected based on the five problem dimensions identified in Section III.

TABLE II. Feature Comparison: Awwazai vs. Commercial Platforms

Feature	Otter.ai	Fireflies.ai	Google Meet Captions	Zoom AI Companion	Awwazai (Ours)
Indian language support	Limited	Limited	Basic	None	Hindi, Marathi, Hinglish, Tamil, Telugu
Mixed-language detection	No	No	No	No	Yes (audio-level)
AI Summarization	Yes	Yes	No	Yes	Yes (multilingual)
Meeting Q&A chat	No	Yes (English)	No	Yes (English)	Yes (multilingual)
Speaker identification	Yes	Yes	No	Yes	Yes (custom labels)
Full-text search	Yes	Yes	No	Limited	Yes (5 fields)
File upload (audio/video)	Yes	Yes	No	No	Yes (background)
Summary language matches speech	No	No	No	No	Yes
Translation support	Yes (paid)	Yes (paid)	Yes	No	Yes (free)
PDF / Transcript export	Yes (paid)	Yes (paid)	No	No	Yes (free)
Open-source / self-host	No	No	No	No	Yes
Pricing	\$8.33–\$20/mo	\$10–\$19/mo	Google account	Zoom subscription	Free (open-source)



As shown in Table II, Awwazai is the only evaluated platform to offer (a) audio-level mixed Indian language detection, (b) summary generation in the detected language, (c) multilingual meeting Q&A, and (d) a fully free offering including PDF export and translation. Commercial platforms consistently restrict advanced features to paid tiers and English language contexts.

VII. CONCLUSION

This paper presented Awwazai, an AI-powered multilingual meeting transcription and summarization platform engineered specifically for Indian language contexts. By combining AssemblyAI's audio-level language detection with Groq-hosted Llama 3.1's multilingual text generation, Awwazai resolves the fundamental limitation of existing commercial tools: their inability to produce meeting summaries and Q&A responses in the actual language spoken during the meeting.

The system's non-blocking upload architecture, parallel chunk summarization, real-time WebSocket transcription, speaker identification with custom labels, cross-meeting full-text search, and zero-cost deployment model collectively position Awwazai as a viable open-source alternative for Indian academic institutions, small businesses, and multilingual teams that are currently underserved by English-first commercial platforms.

Future work will focus on fine-tuning Llama 3.1 on a curated Indian code-switching corpus to improve live transcription accuracy, implementing calendar integration for automated pre-meeting context loading, and extending speaker diarization to support identification from voice embeddings rather than manual labeling.

ACKNOWLEDGMENT

The author acknowledges the open-source communities behind FastAPI, AssemblyAI, Groq, and Tailwind CSS whose tools made this project possible. Special thanks to the faculty supervisors at the Department of Artificial Intelligence and Data Science for their guidance throughout the development of this project.

REFERENCES

- [1] Otter.ai, "Otter.ai: AI Meeting Assistant," <https://otter.ai>, 2024.
- [2] Fireflies.ai, "Fireflies AI Meeting Notetaker," <https://fireflies.ai>, 2024.
- [3] Google, "Live captions in Google Meet," <https://support.google.com/meet>, 2024.
- [4] Zoom Video Communications, "Zoom AI Companion," <https://zoom.us/ai-companion>, 2024.
- [5] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 12449–12460.
- [6] E. Salesky, M. Wiesner, J. Bremerman, R. Cattoni, M. Negri, M. Turchi, D. W. Etter, and M. Post, "Multilingual TEDx corpus for speech recognition and translation," in *Proc. Interspeech*, 2021.
- [7] S. Khare, A. Mittal, and P. Bhattacharyya, "Hindi-English code-switching speech corpus and baselines for ASR," in *Proc. LREC*, 2022.
- [8] OpenAI, "GPT-4 Technical Report," arXiv preprint arXiv:2303.08774, 2023.
- [9] Meta AI, "Llama 3: Open Foundation and Fine-Tuned Chat Models," <https://llama.meta.com>, 2024.
- [10] Groq Inc., "Groq LPU Inference Engine," <https://groq.com>, 2024.

