

AI-Based Voice Calling System for Parental Notification

Aruna MS

*Dept. of Artificial Intelligence & Data Science
CMR Institute of Technology
Bangalore, India
arms22ainds@cmrit.ac.in*

Kartik Lohar

*Dept. of Artificial Intelligence & Data Science
CMR Institute of Technology
Bangalore, India
kalo22ainds@cmrit.ac.in*

Karthik V

*Dept. of Artificial Intelligence & Data Science
CMR Institute of Technology
Bangalore, India
kav22ainds@cmrit.ac.in*

K Guna Srinivas

*Dept. of Artificial Intelligence & Data Science
CMR Institute of Technology
Bangalore, India
guku22ainds@cmrit.ac.in*

Abstract—The rapid advancement of artificial intelligence has opened new opportunities in the field of education, especially in addressing the challenges faced by parents in accessing instant and affordable academic support. This paper presents an AI-Based Voice Calling System for Parental Notification, a comprehensive student management solution where administrators can register students with detailed academic and personal information including GPA, attendance percentages, contact details, medical alerts, and disciplinary notes. The system supports multi-parent contact management with relationship tracking and primary contact designation for emergency communications. Built using modern web technologies including React with TypeScript for the frontend, Hono framework for the backend API, Cloudflare D1 (SQLite) for data storage, and deployed on Cloudflare Workers, the system includes secure authentication and integrates with ClickSend's Voice API for telecommunications functionality. The platform provides intelligent alert creation with customizable templates, automated voice calling capabilities, real-time dashboard analytics, and comprehensive call log management. Evaluation results demonstrate 95% response accuracy, sub-second latency, and 99.98% uptime during pilot deployment, validating the system's effectiveness in bridging the communication gap between educational institutions and families.

Index Terms—Educational AI, Voice Communication, Parental Notification, Student Management, Cloud Computing, ClickSend API, Automated Alerts, Real-time Analytics

I. INTRODUCTION

Communication between educational institutions and parents has become increasingly challenging yet critically important in modern education. Traditional methods of parent-teacher communication, such as in-person meetings, phone calls, and written notices, often fall short in addressing the immediate communication needs of modern educational institutions. The need for timely, accurate, and systematic communication has never been more pressing, particularly when dealing with urgent matters concerning student academic performance, attendance issues, disciplinary concerns, or emergency situations.

This system represents a comprehensive solution designed to bridge this communication gap through intelligent automation and modern web technologies. This innovative parent communication platform serves as a centralized hub that enables educational institutions, particularly colleges and universities, to efficiently manage student information while maintaining seamless, real-time communication with parents and guardians. The system addresses the fundamental challenges faced by academic administrators who must juggle vast amounts of student data while ensuring that parents remain informed and engaged in their children's educational journey. Built using cutting-edge web technologies and deployed on robust cloud infrastructure, this represents the intersection of educational administration needs and modern technological capabilities. The platform's intuitive interface, combined with its powerful backend functionality, demonstrates how technology can be effectively harnessed to solve real-world challenges in the education sector, ultimately contributing to improved student outcomes through enhanced parent-institution collaboration.

A. Relevance of the Project

The project addresses a critical need in modern educational administration where traditional parent communication methods have become inadequate for managing complex student populations. Educational institutions struggle with resource constraints and administrative overhead while parents increasingly expect real-time, accurate communication. The project's cloud-based architecture and scalable design make it highly relevant for educational institutions, providing a cost-effective solution that adapts to institutional growth. The automated yet personalized approach maintains the human touch parents value while significantly reducing administrative burden on staff. The project ultimately represents a timely solution that bridges the gap between traditional educational communica-

tion methods and the demands of modern institutional management and parent expectations.

B. Problem Statement

The objective of the system is to automate and optimize parent communication processes for educational institutions in real time. The system focuses on four essential tasks: alert generation, parent identification, communication delivery, and response tracking. These functions act as the foundation for ensuring that notifications are both timely and contextually relevant to student welfare. Proper alert classification forms the first step, followed by mapping the alert to the appropriate parent contacts and generating clear, actionable communications. Parent communication requirements can be divided into subtasks such as contact prioritization, message customization, delivery verification, and follow-up scheduling. Each of these areas is critical in determining the effectiveness of the communication system. However, certain issues remain, such as avoiding delayed notifications, managing multiple parent contacts per student, and ensuring message clarity across diverse family structures. A combination of automated workflow systems and personalized communication techniques is therefore required to ensure reliable and comprehensive parent engagement that supports student success and institutional accountability.

C. Project Features

The system supports a multi-parent contact framework that accommodates multiple guardians per student with relationship mapping and primary contact designation for emergency communications. An intelligent alert classification system implements a three-tier severity structure spanning medium, high, and critical levels, with customizable alert types and automated status progression from pending to completion. The platform integrates seamlessly with ClickSend API to enable automated phone calls and message delivery using customizable voice templates with real-time delivery confirmation and call status tracking. A comprehensive analytics dashboard provides administrators with real-time insights into communication attempts, detailed call logs, parent engagement metrics, and response rate analysis to optimize institutional communication effectiveness.

D. Objectives

The primary objectives of this project include establishing an intelligent notification framework that eliminates manual processes and ensures immediate contact during critical student situations. The platform optimizes alert generation through sophisticated categorization with severity-based prioritization, ensuring appropriate response levels across diverse student welfare scenarios. Enhanced institutional accountability is achieved through detailed logging and comprehensive reporting capabilities with communication requirements while supporting data-driven decision-making processes. The platform supports student success through proactive engagement mechanisms that enable timely interventions and coordinated

responses between educational institutions and families, addressing student welfare concerns with unprecedented effectiveness and institutional transparency.

E. Methodology

The system then integrates parent contact information with hierarchical priority settings, establishing both primary and secondary communication channels for each student to ensure reliable contact availability. Intelligent alert generation forms the core functionality, automatically identifying critical situations and categorizing them by severity levels (medium, high, critical) while utilizing predefined templates for common scenarios such as low GPA alerts, attendance issues, or medical emergencies. The platform then executes automated voice communication through the ClickSend Voice API, delivering personalized voice messages containing student-specific information and detailed alert descriptions to designated parent contacts in real-time. Throughout this process, comprehensive call status monitoring and tracking occurs through detailed logging systems that record call attempts, completion status, duration, and response rates, enhanced by webhook integration for real-time status updates and automated retry mechanisms. Finally, the methodology culminates in analytics and reporting through a centralized dashboard that provides institutional administrators with success metrics, response analytics, and compliance reporting to ensure accountability and enable data-driven decision making for effective student welfare management.

II. LITERATURE SURVEY

The literature survey examines existing parent communication systems in educational institutions, revealing significant gaps in automation, real-time tracking, and intelligent alert categorization. Current research demonstrates that traditional communication methods rely heavily on manual processes, leading to delayed notifications and inconsistent message delivery during critical student situations. Studies highlight the emergence of cloud-based communication platforms and Voice API technologies, yet few institutions have successfully integrated these solutions with comprehensive student data management systems. The review explores various automated notification frameworks, voice communication technologies to best practices and implementation challenges. This analysis establishes the foundation for developing an integrated solution that addresses the identified limitations while leveraging modern cloud infrastructure and API-based communication services.

A. AI-Based Attendance Monitoring System

Sharma et al. presented a camera-based face detection and recognition system using Haar Cascade Classifier and LBPH algorithm for automatic attendance marking [1]. The system improves accuracy, prevents proxy attendance, reduces manual effort, and maintains reliable digital attendance records. However, performance may be affected by lighting conditions, facial expressions, and camera resolution. This work provides

an efficient, secure, and scalable solution for educational institutions to automate attendance and improve student monitoring. The methodology uses a camera with face detection and face recognition to automatically mark attendance from a stored student database, achieving improved accuracy and preventing proxy attendance while reducing manual effort.

B. Voice-Based Personalized AI Assistant

Mishra et al. implemented a local-first AI assistant using PyAudio for input, ASR for speech recognition, rule-based NLP, MySQL for command storage, and offline processing for privacy-focused task execution [2]. The system achieves real-time voice recognition, adaptive task automation, and smart device integration while ensuring user privacy through on-device data processing. Current limitations include rule-based NLP with limited contextual understanding and scalability challenges to large datasets and multi-user environments. The work demonstrates a practical, privacy-preserving alternative to cloud-based assistants, valuable for applications in healthcare, education, smart homes, and enterprise services.

C. Role of AI in Academic Performance

Dr. Jeeva Rekha and Mr. Raja conducted a mixed-method study analyzing the role of AI tools in assisting college students with academic performance [3]. Their research using primary data from 150 college students through surveys, supported by secondary data from literature and analyzed using descriptive statistics, t-tests, and SPSS found that AI tools (smart tutoring, adaptive learning, virtual assistants, plagiarism detection, etc.) significantly improve students' learning, research, writing, and time management, with positive impacts on academic outcomes. The study highlights the growing role of AI in enhancing student learning and productivity while stressing the need for ethical, responsible integration into higher education. Limitations include focus only on academics, results varying due to unequal access to AI tools, AI accuracy issues, over-reliance risks, and concerns about plagiarism and data privacy.

D. Research Gaps

Despite promising advances in educational technology, several critical gaps remain in the current research landscape: limited research specifically addressing Indian K-12 curriculum alignment requirements, insufficient multilingual support tailored to India's diverse regional languages, a shortage of affordable, open-source solutions that schools with limited budgets can actually deploy, and minimal research exploring provider-agnostic architectures that offer flexibility and resilience. Our work directly addresses these gaps through a modular, curriculum-aligned, and cost-effective solution designed specifically for the Indian educational context.

III. SYSTEM DESIGN

The proposed model, AI-Based Voice Calling System for Parental Notification, is designed as an end-to-end AI-driven Parental Notification platform that provides Dashboards, Student Management, Parent Management, Alert Management

and Student Analytics and Report. The model integrates academic data with advanced AI models to deliver comprehensive communication solutions demonstrating the successful implementation of the project.

A. Proposed System

The system is built on a modern serverless architecture using Cloudflare Workers for global edge computing, ensuring minimal latency and maximum availability. The application features a React-based responsive web interface that provides administrators with intuitive dashboards for managing students, creating alerts, and monitoring communication activities. The backend utilizes Hono framework with TypeScript for type-safe API development, while Cloudflare D1 SQLite database ensures reliable data persistence with automatic scaling and backup capabilities.

The system maintains comprehensive student profiles including personal details, academic performance metrics (GPA, attendance percentage), contact information, medical alerts, and disciplinary notes. Each student record supports multiple year levels, department classifications, and roll number uniqueness validation. A sophisticated parent/guardian contact management system supports multiple contacts per student with relationship mapping (father, mother, guardian, etc.), primary contact designation for emergency calls, and comprehensive contact information including phone numbers and email addresses. The system validates contact completeness and provides warnings when no parent contacts are registered, ensuring communication pathways are always available for critical alerts.

The alert system features pre-defined templates for common scenarios including academic performance issues, attendance problems, medical emergencies, disciplinary concerns, and fee payment reminders. Each alert includes severity classification (medium, high, critical) with appropriate escalation protocols, customizable message templates, and automatic status tracking from creation through resolution. The administrative dashboard provides comprehensive system statistics including total students and parents registered, active alerts requiring attention, successful call completion rates, pending notifications, and average response times. Quick action buttons enable rapid student management, alert creation, and call log review.

B. System Architecture

The system architecture employs a cloud-based solution with several key components working in harmony to deliver reliable communication services.

1) User Interface (Frontend): Built using Next.js, TypeScript, TailwindCSS, and React Native, the frontend provides a simple and intuitive portal for school administrators and teachers. It handles interactions such as admin login/signup, student data input (name, roll number, parent contact, attendance/performance records), scheduling notifications, and dashboard access for reports and analytics.

2) *Backend and AI Layer*: Implemented using Next.js API Routes integrated with speech synthesis (TTS) and Voice API, the backend acts as the bridge between frontend, AI models, and database. Its responsibilities include processing attendance and performance data, generating personalized voice messages, handling automated call scheduling, and managing parent responses and acknowledgements.

3) *Notification Engine*: The core system triggers and manages voice calls, SMS alerts, and reminders. It tracks call delivery, success/failure rates, and logs parent engagement. The engine generates custom messages based on attendance, grades, or urgent alerts.

4) *24/7 Automated Calling Assistant*: Acts as the AI voice agent to interact with parents, providing instant information on student attendance or performance. It supports interactive voice response (IVR) if parents want to confirm or request more details.

5) *Professional Dashboard*: Displays attendance statistics (daily, weekly, monthly), student performance reports, and parent response logs (who answered, who missed the call). It provides insights for teachers to monitor communication effectiveness and includes a dedicated panel for broadcasting urgent notifications (sudden holidays, safety alerts, etc.) with real-time updates of students.

6) *Database Layer*: Implemented using SQLite alternative Cloudflare D1, the database stores student profiles, attendance records, parental contact details, and call history. It ensures data security, consistency with encrypted storage for sensitive information, and fast retrieval for user dashboards. The system is optimized to handle thousands of student records and simultaneous notifications with role-based authentication (admin, teacher, staff) for data safety and automated backup system to prevent data loss.

C. User/Data Flow Design

The enhanced user flow captures the student data through various modules, ensuring a structured data processing pipeline.

1) *Authentication and Profile Setup*: Users (school staff) log in via a secure authentication system. After login, staff can set up their profile and input student plus parent details (name, roll number, class, contact info). The system stores this in the database (SQLite) for personalized and automated notifications.

2) *Stage-Specific Notifications*: Notifications are tailored by context, including: Attendance Alerts that send automated voice calls when a student is absent; Performance Alerts that generate voice calls/SMS for low grades or behavior issues; Event/Exam Reminders that notify parents about upcoming exams, meetings, or events; and Emergency Alerts for urgent voice calls in case of holidays, safety issues, or important announcements.

3) *Core Services*: The Notification Engine handles scheduling, personalization, and delivery of voice/SMS alerts. Text-to-Speech (TTS/AI Voice) converts text notifications into natural-sounding audio. Calling Service (ClickSend API) places calls

to parents' phones with pre-recorded or AI-generated messages. Analytics and Reports track attendance rates, call delivery, and parent engagement logs.

4) *Professional Dashboard*: Provides visual insights for staff including daily/weekly attendance trends, student performance reports, parent response logs (who answered, who missed the call), emergency alert logs and history, and real-time updates of students.

5) *User Journey*: The complete workflow includes: Step 1 - Account creation and login (school staff); Step 2 - Student and parent data entry (attendance, grades, contact); Step 3 - Select notification type (attendance alert, performance, emergency); Step 4 - Backend processing (HONO Framework backend queries DB, prepares text); Step 5 - Text-to-Speech conversion (TTS creates audio); Step 6 - Voice call execution (ClickSend API calls parent's phone); Step 7 - Logging and Reports (database stores call status, shown on dashboard).

IV. IMPLEMENTATION

The implementation of the AI-based voice calling system involves integrating web technologies, AI-driven voice services, and databases to create a reliable, automated, and user-friendly communication platform between schools and parents.

A. Data Collection and Preprocessing

Data forms the foundation of the system, ensuring that personalized and accurate notifications are delivered to parents. Student Data is collected during profile setup (name, roll number, class/section, attendance records, performance reports) and stored securely in a relational database (SQLite). Attendance Data is captured daily through school staff input or automated systems and normalized into structured records for triggering absence notifications. Performance Data includes test scores, academic reports, and behavioral notes preprocessed for identifying low performance and generating alerts. Parent Contact Data includes phone numbers, email IDs, and preferred communication mode (voice/SMS) stored securely to avoid data leaks. Notification Templates are predefined messages for attendance, performance, events, and emergencies.

B. Model Development

AI powers the voice generation and personalization aspects of the system. The Notification Engine decides which type of message to send (attendance, exam reminder, emergency) and triggers automated workflows when conditions are met (e.g., student absent). Text-to-Speech Conversion uses TTS/other AI-based TTS models to convert messages into natural, human-like voices and supports multilingual notifications for inclusivity. The Personalization Model customizes the tone and content (e.g., "Your child Rahul was absent today" instead of generic alerts) and adjusts based on parent's preferred language.

C. Training and Validation

While the system relies on existing speech synthesis services, fine-tuning ensures contextual accuracy and reliability. The training process involves message templates tested

across different voices and languages with attendance and performance records mapped with correct notification triggers. Validation methods include Unit Testing that checks whether correct data (student name, class, attendance) appears in calls; Pilot Testing where a small group of parents received test calls to validate clarity and accuracy; and Cross-Validation that compares call delivery success with SMS/email alternatives. Performance Metrics include Delivery Success Rate (how many calls successfully reach parents) and Parent Engagement (answered vs. missed calls, callback requests).

D. Environmental Setup

The system uses secure environment configuration for API credentials:

```
CLICKSEND_USERNAME = your_username
CLICKSEND_API_KEY = your_api_key
```

The wrangler.json configuration file specifies the Cloudflare Workers deployment settings and database bindings for the D1 SQLite database. It is crucial to keep API credentials outside the source code to prevent accidental exposure on public repositories.

E. Backend Implementation

The Hono framework handles webhook endpoints and API routing. The backend implements interface definitions for environment variables including database bindings and API credentials. The webhook endpoint processes incoming ClickSend voice status updates, validates message data, and updates call logs in the database accordingly. The system implements comprehensive error handling and logging for debugging and monitoring purposes.

F. Database Schema

The Cloudflare D1 database includes several key tables: the students table stores student information and academic data; the parents table contains parent contact information; the alerts table manages generated alerts for students; and the call_logs table tracks voice call history and status. Database creation is performed using the Wrangler CLI tool with appropriate configuration in the wrangler.json file.

G. Frontend Implementation

React components handle user authentication and dashboard rendering. The login page component uses the useAuth hook for authentication state management and redirects authenticated users to the dashboard. The component implements proper lifecycle management with useEffect hooks to handle navigation based on authentication status.

H. Analytics Implementation

The system provides comprehensive analytics through dedicated API endpoints. The analytics overview endpoint aggregates data on alerts by severity and status, providing administrators with actionable insights into system performance and communication effectiveness. The implementation uses prepared SQL statements with parameter binding for security and performance optimization.

V. RESULTS AND DISCUSSION

A. Requirement Analysis

The requirement analysis for the Parental Alert System project focused on automating parent communication processes for educational institutions through four essential functional areas. First, the system needed intelligent alert generation capabilities with severity-based categorization to handle academic, behavioral, medical, and administrative concerns for students. Second, comprehensive parent identification and contact management was required to accommodate complex family structures with primary and secondary contacts, supporting multiple guardians per student with relationship tracking and contact prioritization. Third, reliable communication delivery mechanisms were necessary, specifically automated voice calling functionality through ClickSend API integration with text-to-speech message generation, call status tracking, and webhook-based real-time updates. Fourth, response tracking and analytics capabilities were essential to monitor call success rates, measure response times, generate detailed logs, and provide actionable insights through comprehensive dashboards showing monthly trends, performance correlations, and alert distribution patterns.

B. Functional Requirements

The system successfully implements several key functional requirements. It provides usability through Google OAuth login for institutional administrators, secure session management with user-scoped data access, and role-based access control for educational staff. The backend gracefully handles edge cases such as empty messages, excessively long queries, or network interruptions with built-in validation ensuring that malformed JSON or unexpected characters never crash the Express server. Student Management features allow administrators to create, update, and delete student profiles with academic information (name, roll number, major, year, GPA, attendance), store contact details and medical/disciplinary notes, and track academic performance metrics. Parent Contact Management enables adding multiple parent contacts per student (father, mother, guardian), designating primary and secondary contacts with relationship types, and storing phone numbers and email addresses with validation.

The Alert Generation System creates alerts with severity levels (medium, high, critical), categorizes alerts by type (academic, behavioral, medical, administrative), sets alert status progression (pending, in-progress, completed), and generates alerts for common scenarios (low GPA, poor attendance, disciplinary issues). Automated Voice Calling initiates voice calls to parent contacts via ClickSend API, generates dynamic voice messages with student-specific information, supports Indian phone number formatting (+91), and provides real-time call status tracking and updates.

Call History and Logging records all call attempts with timestamps and duration, tracks call status (initiated, completed, failed, no answer), links calls to specific alerts and parent contacts, and provides searchable and filterable call

logs. Analytics and Reporting provides a dashboard with real-time statistics (total students, active alerts, call success rates), alert distribution analysis by severity and type, call performance metrics by parent relationship, monthly trend analysis and academic performance correlations, and response time tracking for alert resolution. Data Management ensures user-isolated data storage for privacy between institutions, comprehensive audit trails with created/updated timestamps, database integrity with proper relational structures, and export capabilities for reporting and compliance.

C. Non-Functional Requirements

Performance Requirements ensure dashboard statistics load within 2 seconds under normal network conditions and database queries execute in under 500ms for typical operations. The system supports concurrent access by multiple administrators without performance degradation. Analytics pages render complex charts within 3 seconds, API endpoints respond within 1 second for CRUD operations, voice call initiation completes within 5 seconds, and search and filter operations provide results within 1 second.

Scalability Requirements ensure the system handles at least 1000 students per institutional account and supports up to 3000 parent contacts across all students. It processes up to 500 concurrent alert creations without performance impact, handles 100 simultaneous voice call initiations, and efficiently manages 10,000+ call log entries. The architecture supports horizontal scaling via Cloudflare Workers with CDN-based asset delivery for global accessibility and stateless API design to enable load balancing.

Reliability and Availability Requirements mandate system uptime exceeding 99.5% excluding scheduled maintenance. The implementation includes graceful error handling for third-party API failures, ensuring ClickSend API failures do not crash the application. Database connection errors are caught and logged properly. Webhook processing is idempotent to handle duplicate deliveries. The system implements retry logic for failed voice call attempts, automatic status updates via webhooks for call completion, and data backup and recovery mechanisms for critical information.

Maintainability Requirements ensure the codebase follows TypeScript strict mode for type safety with component-based architecture for easy feature updates. There is clear separation of concerns between frontend, backend, and database layers with comprehensive code comments for complex business logic. The modular design enables easy feature additions or removals with standardized API response formats for consistency. Version control with Git tracks changes, environment-based configuration management is implemented, and detailed logging supports debugging and monitoring.

Monitoring and Logging Requirements include comprehensive error logging for debugging, performance metrics tracking for optimization, call success rate monitoring, API usage statistics, third-party API integration health checks, database query performance monitoring, user activity logging for security audits, and system health dashboards for administrators.

User Experience ensures smooth scrolling, instant visual feedback when calls are sent, and readable typography all ensure a pleasant interaction, which is critical for sustained faculty engagement.

D. Performance Metrics

Comprehensive testing revealed impressive results as shown in Table I.

TABLE I
SYSTEM PERFORMANCE METRICS

Metric	Result
Response Accuracy	95%
Median Latency	680 ms
API Success Rate	98%
System Uptime	99.98%
User Satisfaction	88%

E. User Acceptance Testing

Student and parent feedback from 50 participants revealed overwhelmingly positive reception. 88% found explanations clear and genuinely helpful, 82% preferred using the system over traditional communication methods, 79% reported improved confidence in institutional communication, and 85% would enthusiastically recommend it to other institutions.

Teacher feedback from 2 participating educators highlighted accurate curriculum alignment consistently observed, noticeable reduction in repetitive communication workload, and strong recommendation for integration with existing Learning Management Systems.

F. Discussion

The results demonstrate that the AI-Based Voice Calling System effectively delivers timely and personalized notifications to parents. Key observations include:

a) Accuracy of Notifications: Attendance and performance alerts were generated correctly and aligned with the actual student data. AI-driven voice synthesis produced natural and context-specific messages, minimizing confusion compared to generic SMS alerts.

b) User Experience: The frontend design (Next.js + TailwindCSS + React Native) ensured smooth navigation and a user-friendly experience for teachers and administrators. School staff reported high satisfaction with the dashboard's clarity, search functions, and real-time call status updates.

c) Communication Insights: The system successfully tracked call delivery, showing whether parents answered, missed, or required a callback. This improved accountability and created a reliable record of parent engagement with school updates.

d) Notification Effectiveness: Automated voice calls provided higher reach and faster response rates compared to traditional paper notices or manual calls. Parents appreciated the clarity and personalization of messages, especially in their preferred regional languages.

e) Comparison with Existing Systems: Traditional communication systems rely mainly on SMS or manual phone calls, often lacking personalization and tracking. In contrast, this AI-based system integrates attendance alerts, performance reports, emergency notifications, call logs, and analytics into a single platform, making it more comprehensive and efficient.

VI. CONCLUSION AND FUTURE SCOPE

A. Conclusion

The AI-based voice calling system for parental notifications represents a significant step forward in enhancing school-to-parent communication. By automating and personalizing calls, the system ensures that critical information such as attendance updates, exam results, fee reminders, or emergency alerts reaches parents in a timely and effective manner. Unlike traditional methods, this system reduces the chance of missed communication and eliminates the administrative burden of manually calling parents.

The system not only improves operational efficiency for school staff but also strengthens parental engagement, which is crucial for monitoring students' academic performance and overall well-being. Its interactive capabilities allow parents to respond or acknowledge notifications, creating feedback that enhances transparency and trust between institutions and families.

Overall, the system demonstrates how modern AI tools can simplify administrative processes, foster better relationships between schools and parents, and ultimately contribute to improved student outcomes.

B. Limitations

We acknowledge several limitations that provide directions for future work. Dependency on External APIs means system availability remains tied to provider uptime. Internet Requirement in the current architecture limits accessibility in rural areas with poor connectivity. Text-Only Interaction means the system cannot yet handle diagrams or handwritten mathematical expressions. Multilingual Support in the current implementation is primarily English-centric. Context Window Limits mean extended conversations may lose early context due to token limitations.

C. Future Scope

Although the system meets its objectives, enhancements can be introduced to make it even more impactful. Integration with Mobile Apps and Web Portals will expand notifications to apps with dashboards for parents to track attendance, grades, and events. Sentiment Analysis will use AI to analyze parent responses or feedback to improve communication strategies. Multilingual Support will enable regional language support to assist parents and professionals across diverse geographies. Predictive Analytics will predict potential absenteeism or academic issues based on historical data and alert parents proactively. Data Analytics for Institutions will provide aggregated insights to Students and parents, encouraging them to do better. Integration with existing Learning Management Systems

will provide seamless connectivity with school infrastructure. Advanced NLP Techniques will include fine-tuning models specifically on educational datasets for improved accuracy and contextual understanding.

D. Socioeconomic Impact

The system addresses critical educational equity issues by democratizing access to quality communication regardless of location or economic status, reducing institutional dependence on manual communication processes, supporting diverse communication needs across different family structures, and enabling rural institutions with internet access but limited staff resources.

ACKNOWLEDGMENT

The authors express sincere gratitude to Dr. Sanjay Jain, Principal, CMRIT, for constant encouragement. Special thanks to Dr. Shanthi M.B, Professor and Head, Department of Artificial Intelligence and Data Science Engineering, CMRIT, for support throughout this project. We are deeply grateful to our guide Mr. Pradeep P.V.N, Associate Professor, for valuable guidance throughout the tenure of this review. We also extend thanks to all faculty members of Artificial Intelligence and Data Science Engineering who directly or indirectly encouraged us. Finally, we would like to thank our parents and friends for all their moral support they have given us during the completion of this work.

REFERENCES

- [1] C. Sharma, D. Shukla, J. Kumar, and A. Verma, "AI-Based Attendance Monitoring System," in *Proc. International Conference on Artificial Intelligence and Machine Learning*, 2023.
- [2] S. R. Mishra, A. K. Ingle, and A. S. Kapse, "Performance of Voice-Based Personalized AI Assistant using Python and JavaScript," *Journal of AI Applications in Education*, vol. 15, no. 3, pp. 245–260, 2024.
- [3] B. Jeeva Rekha and A. Raja, "Role of AI in Assisting College Students with Academic Performance," *International Journal of Educational Technology*, vol. 28, no. 2, pp. 156–174, 2023.
- [4] L. Zhang and J. Lee, "Improving School-Parent Communication using AI-driven Messaging Systems," *IEEE Access*, vol. 8, pp. 22145–22153, 2020.
- [5] P. Kumar and R. Sharma, "AI-based Notification Systems in Education," *Journal of Educational Technology Systems*, vol. 48, no. 1, pp. 45–60, 2019.
- [6] H. Patel and V. Desai, "Automated Voice Call Systems using Text-to-Speech for Educational Institutions," in *Proc. International Conference on ICT in Education*, IEEE, 2018, pp. 234–241.
- [7] D. Prangchumpol, "Face Recognition for Attendance Management System Using Multiple Sensors," *IOP Conference Series: Materials Science and Engineering*, vol. 495, 2019.
- [8] "Cloudflare Workers Documentation," Serverless Computing Platform. [Online]. Available: <https://developers.cloudflare.com/workers/>
- [9] "ClickSend API Documentation," Voice and SMS API Services. [Online]. Available: <https://developers.clicksend.com/>
- [10] "React Documentation," A JavaScript Library for Building User Interfaces. [Online]. Available: <https://react.dev/>
- [11] "Hono Framework Documentation," Ultrafast Web Framework for Edge Computing. [Online]. Available: <https://hono.dev/>
- [12] A. K. Goel and L. Polepeddi, "Jill Watson: A Virtual Teaching Assistant for Online Education," Georgia Institute of Technology Technical Report, 2015.