IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Rural Gyan Platform: An AI-Integrated Full-Stack Virtual Learning System for Rural India

Neha Gaikwad[1], Meghna Prathipati[2], Saylee Shelar[3], Dr. Sandeep Kulkarni[4]
B.Tech, Student[1], Department of Computer Science
B.Tech, Student[2], Department of Computer Science
B.Tech, Student[3], Department of Computer Science
Assistant Professor[4], Department of Computer Science, Pune, Maharashtra
Ajeenkya DY Patil University
Lohgaon, Pune, Maharashtra, India

Abstract: While technological developments continue to advance, the divide in the quality of education remains a problem in developing countries. The problem is not only the infrastructure in the developing country, but the language barrier as well. Therefore, this essay will analyze the architecture of the "Rural Gyan Platform" educational solution that is created to tackle the existing issues in education in developing countries. To begin with, this platform utilizes the latest technology stack (React.js – Client-side; Node.js, Express.js – Server; MongoDB – data persistence; Socket.io – real-time communications), which enables the platform to provide a full-stack educational software product. This particular platform is meant to resolve some of the unique issues that arise in the context of developing countries, namely, low-speed internet, language issues, and lack of interactive digital learning tools. As for the architecture of Rural Gyan, this educational software uses a highly developed role-based system, providing Administrators, Teachers, and Students with their personal dashboards. What makes this project stand out among other similar projects is the possibility of creating the virtual classroom through peer-to-peer connection using WebRTC, avoiding all the potential issues related to overloaded relay servers because of insufficient bandwidth. Another thing worth noting is the fact that the use of the Google's Gemini large language model allows introducing the multimodal AI tutor. Lastly, it should be said that the platform avoids language barriers, providing built-in bilingual support (react-i18next and live translations from Google).

Keywords: WebRTC, Artificial Intelligence, Educational Technology, Socket.io, React.js, Full-Stack Development, Peer-to-Peer Networks.

I. Introduction

The process of democratization of education using digital mediums has brought about significant changes to pedagogy worldwide. Nonetheless, the nature of this digital evolution is disproportionately represented. The issue becomes more pronounced when looking at India. Given the rural populace of over 650 million people, the technological presence is suboptimal. According to the latest statistics, broadband penetration in rural areas stands around 27% compared to much better numbers in urban areas. In addition, the issue of language barrier is quite problematic. Since Hindi and regional languages serve as the primary languages used in communicating and teaching in rural India, 92% of all digital educational content is still in English.

The process of evaluation of existing market products leads to the identification of the following major architectural limitations:

Google Classroom: The primary function served by Google Classroom is that of an assignment and documents management software tool. Does not come with any built-in WebRTC functionality for video conferencing and whiteboard features simultaneously.

Moodle: Moodle has traditionally been designed as a monolithic PHP application and thus relies on several plug-ins for synchronous communication, hence, leading to excessive bloated installations and inefficient performance in live application execution.

Zoom Education: Best suited for video delivery yet operates on a system where everything goes through the relay servers. The system of relaying information does not make for effective performance since it demands more baseline computer bandwidth without using peer-to-peer mesh network connectivity.

Khan Academy: Offers excellent content delivery in module fashion but works in an asynchronous mode without immediate feedback from educators to pupils

In order to overcome the gap mentioned above, the Rural Gyan Platform has been designed. Our major contributions can be highlighted through the following two contributions: first, design of the overall architecture of the system; second, overall architecture design of a complete educational platform utilizing state-of-the-art technologies such as real time WebRTC, Collaborative Whiteboard, Live Bilingual captions (English-Hindi), and AI Tutoring.

II. LITERATURE REVIEW

Artificial Intelligence in Education:

Utilization of Artificial Intelligence (AI) technology for educational purposes has increased within the last two decades. The authors carried out a bibliometric analysis of the field of study of Artificial Intelligence in Education (AIEd) entitled "Two Decades of Artificial Intelligence in Education: Contributors, Collaborations, Research Topics, Challenges, and Future Directions." The authors observed that there is an increasing trend for a changeover from rule-based intelligent tutoring system to data-based adaptive learning system. Some of the key research areas include personalization, learner analytics, automated assessment, and recommendation systems. They also observed some critical challenges including poor infrastructural support, issues related to privacy, bias in algorithms, and unequal distribution of technology especially in underdeveloped and rural areas [1].

Likewise, the paper titled "Evolution and Revolution in Artificial Intelligence in Education" details how the educational technologies have evolved from being conventional forms of e-learning technologies into intelligent and contextual forms of learning technologies. Based on the paper, the field of education has been revolutionized by the use of artificial intelligence through such things as adaptive tutoring, intelligent content development, learning analytics, and feedback mechanisms. However, according to the paper, most of the present-day education systems remain fragmented and have failed to successfully integrate communication technology into artificial intelligence [2].

A more general view is discussed in "Artificial Intelligence in Education: A Review," in which the researchers identified five main uses of AI technology in education: intelligent tutoring systems, conversation agents, automatic assessment, recommendation systems, and learning analytics. However, the review highlights some significant constraints associated with the technology, including its inability to support multiple languages, its limited contextual understanding, and its restricted use in areas with poor internet connectivity [3].

WebRTC and Real-Time Communication:

The communication mechanism of the proposed platform is highly backed by WebRTC-related studies. In particular, in "WebRTC Testing: Challenges and Practical Solutions," the authors discuss the challenges associated with developing WebRTC-based software products. They highlight challenges including browser compatibility, unstable network, signaling process difficulties, NAT traversal, and managing latency. These issues can be managed with the help of STUN/TURN servers, adaptive bitrate technology, and testing automation tools to maintain peer-to-peer communication [4].

In the paper titled "Real-Time Virtual Classroom Platform: Integrating React, Firebase, WebRTC, and Socket.io," the researchers introduce a virtual classroom architecture involving React on the frontend side, WebRTC for peer-to-peer connection, Socket.io for signaling, and Firebase for synchronization. The findings show that integrating these tools can ensure efficient real-time video communication and collaboration in the virtual classrooms. However, the lack of intelligent tutoring functionality and multilingual features is an issue that should be addressed in a rural environment [5].

In addition to this, there is a piece entitled "WebRTC Technology Overview and Signaling Solution Design and Implementation" which has elaborated on WebRTC technologies in details like media capture, SDP, ICE candidates, STUN/TURN, and signaling. The researchers have found out that WebRTC is an effective and economical way of conducting online interactions due to lesser dependence on central servers. It has been pointed out that signaling technologies using WebSocket or Socket.io need to be developed in order to make peer connections secure [6].

Peer-to-Peer Networking:

The general concept of decentralized communication has already been discussed in the paper entitled "A Survey of Peer-to-Peer Networks." In this research, different types of peer-to-peer structures were studied, such as structured and unstructured peer-to-peer, and were rated based on factors like scalability, reliability, and fault tolerance. It was found out that peer-to-peer communication reduces dependency on any centralized system and still performs well even in poorly connected networks [7]. Hence, peer-to-peer communication can be very useful in remote educational communities since centralized video communication systems would fail due to inadequate bandwidth.

Research Gap:

Based on the findings from the literature review, it can be seen that either the existing education portals have no real-time collaborative features and provide AI-based tutoring or use WebRTC technologies to design the online classroom without any AI-based tutoring and multilingual support. There is no consideration for the special problems associated with the rural areas, like poor Internet connectivity, language problems, and lack of infrastructure. There is definitely a need for an all-inclusive tutoring, WebRTC-based peer-to-peer classes, bilingual interaction, and scalable real-time collaboration in one solution.

III. RESEARCH METHODOLOGY

The Rural Gyan Platform is designed based on a streamlined three-tier architecture that ensures complete decoupling between presentation layer, business logic layer, and database layer.

Architectural Topology:

Client Tier: The SPA built using React.js is referred to as the Client Tier. This is helpful in managing the states of the application and efficient render processes on the browser. Application Tier: The Application Tier plays the role of being the RESTful API server, and it is built using Node.js and Express.js. Data Tier: The Data Tier uses MongoDB Atlas – a NoSQL document database.

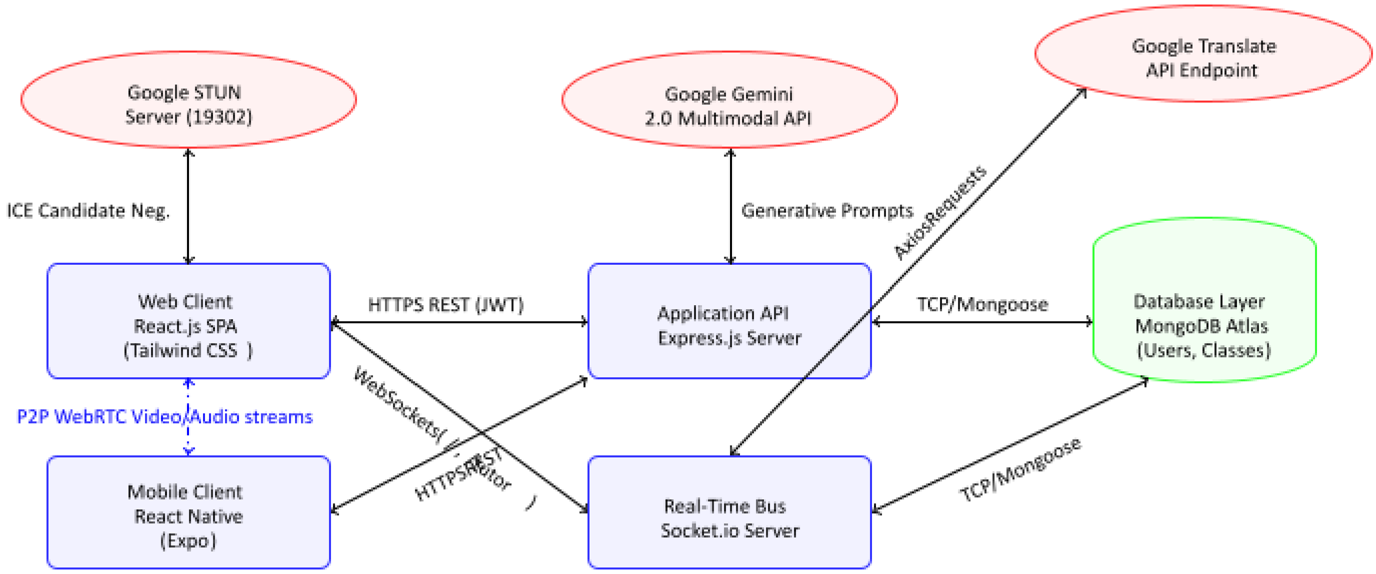

fig. 1. comprehensive full-stack architecture depicting relationships across isomorphic client ecosystems, decoupled express.js/socket.io service tiers, atlas nosql persistence, and integrated external third-party apis.

Real-time Communication Infrastructure:

Socket.io acts as an abstraction layer for WebSocket protocol to facilitate real-time synchronization among the clients. In order to separate the event flow in the system, the namespaces are separated. Though there is a namespace that handles all events concerning virtual class signaling, chatting, and whiteboarding, there is also a namespace known as /tutor for handling all events concerning AI tutor queries.

Security and Middleware Configuration:

For TLS (Transport Layer Security) and authentication, it would be achieved using JSON Web Tokens (JWT). For the infrastructure side, a two-token approach is required, including the Access Token, which is short-lived and encrypted, and the Refresh Token, which is long-lived and saved on the local end in HttpOnly cookies. Passwords will also be hashed with bcrypt with a work factor (salts rounds) of 12. Access to the APIs will be restricted through RBAC middleware, including cryptographically signed requests and roles (admin, teacher, student).

Some other security controls in terms of the infrastructural aspect include limiting the number of requests, up to a maximum of 100 requests per 15 minutes per IP address using the parameterized express-rate-limit module, to guard against brute force and DDoS attacks. Additionally, using Helmet.js, which allows for protection against XSS attacks and clickjacking using security headers, as well as CORS whitelisting to limit domain access to specific URLs only.

IV. THEORETICAL FRAMEWORKS

Role-Based Dashboard System:

Interface design needs special contexts that are limited by the user's permission level, which can be accomplished by using the Tailwind CSS language in an immersive way, whereby the design language will have its aesthetics centered around a dark theme inspired by cyberpunk.

Admin Dashboard:

The administrators enjoy overarching mutative powers. The dashboard provides all the macro level analytics, which include analyzing the MongoDB schemas to generate quantitative data (teachers enrolled, students enrolled, simultaneous sessions held, and anomalies detected from the logs). It provides full create-read-update-and soft-delete (CRUD) operations on a heuristic basis of toggling status (toggle the isActive boolean schema attribute without making any destructive operations) of Teacher and Student objects.

fig 2: admin dashboard interface showing class statistics, teacher management, and logs and reports.

Teacher Dashboard:

Teacher Portal plays a central role in all the academic activities. Material upload, which is done asynchronously, can be done through the Material Management system, whereas the scheduling, starting, and forced termination of Live Virtual Classes happen through the Class Management system. The Quiz Management system allows teachers to conduct quizzes that can either be Classical Multiple-Choice Questions or Algorithmic Coding questions with parameters. Administrative systems also include Subject Index Allocated to teachers, Performance Analysis Arrays that give regular updates on the performance of the students, and Weekly Timetable Chronology.

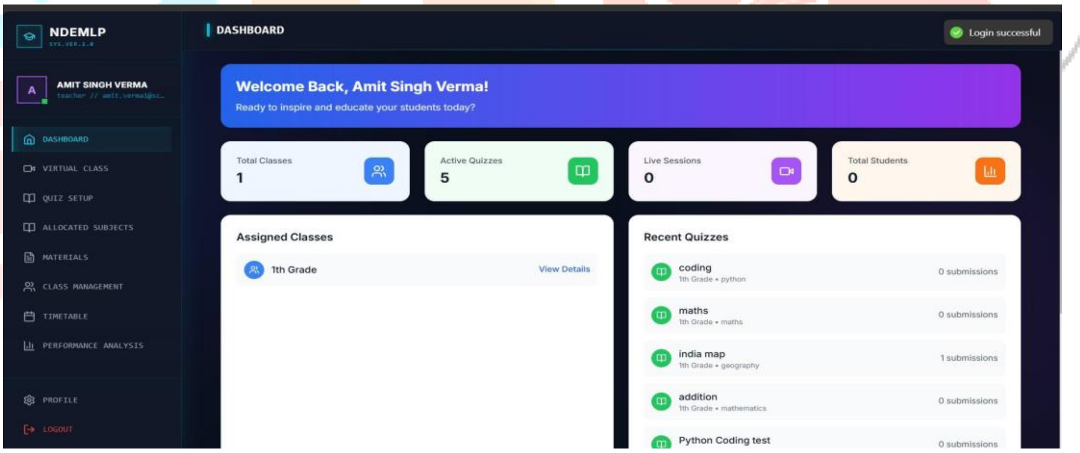

fig 3: teacher dashboard interface showing class statistics, assigned classes, and recent quizzes.

Student Dashboard:

Students work in an organized and streamlined experiential environment. Direct integration for getting their ongoing Virtual Class sessions and direct join links for the class sessions is also offered. There is also the provision of accessing the deposited learning materials and quizzes with the help of time frames in which quiz taking occurs, as well as Timetabling and Virtual Code Editor.

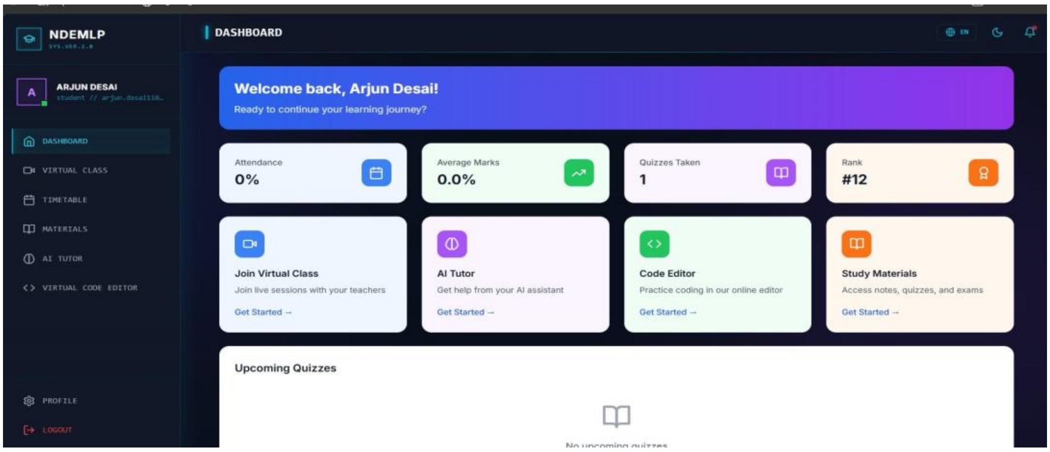

fig 4: student dashboard interface showing virtual class, ai tutor, and code editor.

Virtual Classroom Subsystem (Core Innovation):

Inadequate bandwidth could be remedied by implementing the WebRTC Local Routing approach which will facilitate skipping the SFU and MCU servers which were found to be ineffective previously when deployed in low bandwidth conditions.

WebRTC Peer-to-Peer implementation:

The model is set up using the full-mesh network design of peer-to-peer communication in the WebRTC protocol family with the use of the wrapper library simple-peer. Negotiation is accomplished using the NAT traversal approach with the assistance of the TRUN server. The dependable STUN server (stun.l.google.com:19302, stun1, stun2) works flawlessly in asynchronous candidate matching.

Socket.io Signaling Mechanism:

The session initiation process is initiated by utilizing Socket.io to communicate the signaling handshake request. The notable events include:

join-virtual-class: Transmits the join event command along with its required parameters.

video-offer & video-answer: Communicates SDP information.

ice-candidate: Transmits ICE candidates.

leave-virtual-class & participant-left: Eliminates peers after leaving the call.

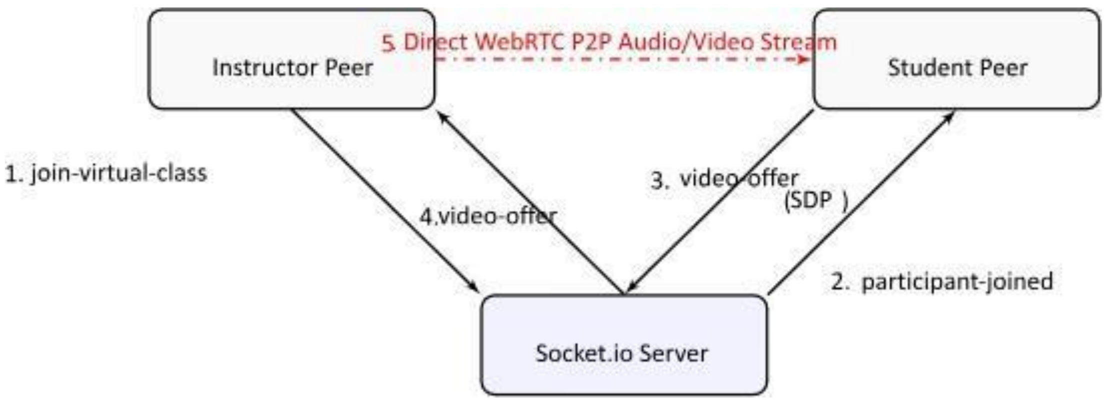

fig 5: webrtc session negotiation and peer-to-peer signaling logic over localized socket.io namespaces illustrating optimal connection routing.

Screen Sharing and Live Captions:

The feature of screen sharing allows for the dynamic replacement of tracks without requiring the renegotiation of session on all the existing peer to peer connections. Voice recognition technology uses the webkitSpeechRecognition API of the Web Speech API; English audio data is captured from the local microphone buffer, and an HTTP POST request is made to the Google Translation API services available for free at translate.googleapis.com/translate_a/. The output is rendered in Hindi text and is published to the room via the caption-update socket event.

Telemetry, Attendance, and Lifecycle Controls

The Lifecycle status rules used in VirtualClass Mongoose are governed by the Enum mapping (scheduled, live, ended, cancelled). The concurrency is achieved through auto-generated cryptography in meeting id which are securely linked with a unique meetingPassword per session. Authoritative control is given to the Instructor using socket payload for controlling certain events in targetted socketIds (for instance, instructor-mute-instructor video off). Hand raise works based on simple Boolean emitter events (raise hand, lower all hands). Automation of attendance happens uniquely through attendance/mark API triggered on join and periodically thereafter till end class broadcast.

Collaborative Whiteboard:

The integration features an HTML5 <canvas> synchronized globally via discrete coordinate transmission paths. All drawing events conditionally execute behind an isTeacher middleware conditional ensuring unidirectional graphic flow. Toolsets encompass variable vector mechanisms (pen, 7color palettes, defined 1-10px line radii) and eraser functionality inherently achieving non-destructive removal utilizing native Canvas destination-out global composite operational flags. Socket telemetry limits to whiteboard-draw (encoding Cartesian *x*,*y* sets, hex colors, and coordinate boundaries) and global wiping arrays. Static exportation outputs raw Base64 image encoding dynamically via canvas.toDataURL() routed safely into standard anchor download elements. Client perspectives are implicitly locked within a ”View Only - Teacher is drawing” localized overlay constraint.

AI TUTOR SYSTEM:

The personalized digital process is characterized by integration with the powerful Google Gemini 2.0 Flash multimodal language model (gemini-2.0-flash) that works by orchestrating the use of @google/generative-ai SDK inside Node.js.

Architecture and Namespace:

Live conversation must be conducted within the isolated /tutor Socket.io namespace so that it is separate from the main WebRTC traffic using the same JWT authorization arrays.

Multimodal Input Capabilities:

Not only does the interface allow plain textual input arrays; students can engage in multidimensional conversations. PDFs are processed using the built-in pdf-parse parser, and all string content is extracted. Word documents (.docx) require binary XML extraction logic created using mammoth, while images undergo OCR treatment by using the Tesseract.js package for extracting strings to process.

Translation Pipeline and Logic Flow:

Translation is done asynchronously throughout the live interaction process. Inputs, which are specifically designed to use native Hindi language, require preprocessing by hitting the Google Translate API normalizing raw inputs into systemic English structures. Then, the string of the English parameters requests the Gemini model for an answer. Then, the answer is translated backwards through the same pipeline.

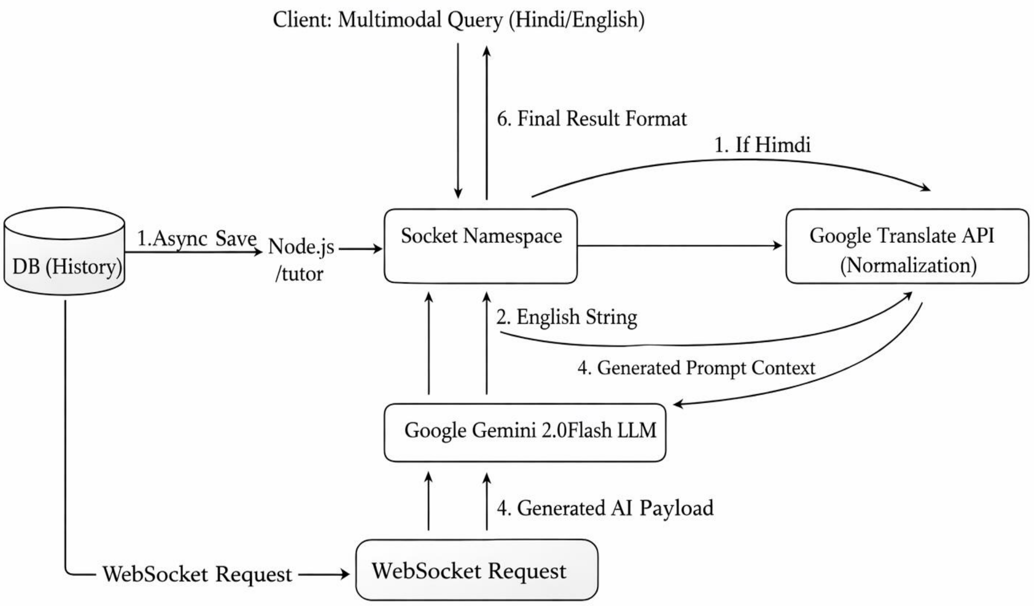

fig. 6:. multimodal ai tutor data pipeline featuring asymmetrical translation processing hooks to normalize non-english queries prior to llm evaluation, maintaining localized high-performance responses.

Frontend Interactions:

Clients interact employing file limits optimized to 10MB capacities. Direct Voice Input leverages the native MediaRecorder API producing localized binary blobs sent strictly as transcription buffers. Responses execute within comprehensive paginated local history tracking architectures (50 queries bound by page) indexed cleanly on the MongoDB AiChatHistory dimensional model. UX optimizations include thumbs-up/down qualitative reactions, instantaneous copy-to-clipboard functionalities, fluid fullscreen expanding wrappers, and visual typing manifestation indicators masking query latency optimally.

QUIZ MANAGEMENT SYSTEM:

The remote assessment system supports various test types, depending on their format and design in a well-organized manner.This system provides the capability to design various kinds of questions, for instance, MCQs and coding questions with the language specification of JavaScript, Python, Java, C++. Questions in the test are separated into categories, where all the questions have clear mappings within the defined section. Teachers have full control on how the exam should be conducted. Teachers will be allowed to assign the total marks on the test, assign weights of different questions, and define the exam duration. An exam has a well-defined starting and finishing time based on teachers' instructions.

This system works predictably and consistently when conducting an examination remotely. MCQ questions have instant evaluation upon submission. Answers provided by students will be evaluated using automated testing and results will be generated immediately. These results include total marks, percentage and total time spent on the exam. Coding questions are tested by providing specific input/output requirements for each student's code. The program written by students will be executed in a secure environment and evaluated accordingly.

Bilingual Support:

Bridging fundamental language disparities within regional localities functions directly utilizing react-i18next architecture. The framework fundamentally sidesteps performance degradation regarding asynchronous external HTTP translation downloads by storing complex hierarchical JSON dictionary mappings internally bounding English and Hindi structural keys directly within memory maps during compilation cycles. More than 80 independent functional translation maps encapsulate every critical node present actively on the DOM encompassing Auth patterns, discrete Dashboard menus, Virtual Class notifications, AI interfaces, and Profile configurations. A persistent, globally scoped Language

Switcher governs current state parameters mutating context cleanly enforcing Hindi logic instantly across parameters spanning navigational targets natively to the overarching institutional tagline translated perfectly to " ".

SECURITY IMPLEMENTATION:

The password mechanism uses hashing by bcryptjs, where salting is performed using a cost of 12. This arrangement ensures protection against quick brute force and rainbow table attacks. The routing function always ensures that input validation is clearly defined, where express-validator filters primitive JSON inputs to safeguard against injection attacks such as NoSQL injections before any action from the controllers. Database changes follow the Soft Deletion principle, where a boolean isActive flag is set to ensure that history is maintained while avoiding data destruction.

Comparative Analysis:

Rural Gyan successfully aggregates sophisticated features dispersed generally across disjointed solutions into one singular context optimized for low-bandwidth implementation factors natively serving Indian operational structures.

table 1: comparative analysis of rural gyan with existing platforms

SYSTEM CAPABILITIES COMPARATIVE ANALYSIS

Capabilities	Rural Gyan	G-Classroom	Moodle	Zoom Edu	Khan Acad
WebRTC P2P Data	Yes	No	No	No	No
Socket Whiteboard	Yes	No	Optional	Yes	No
AI Tutor (Gemini)	Yes	No	No	No	No
Live Bilingual Caps	Yes	No	No	Paid	No
Teacher Mute/Vid	Yes	N/A	N/A	Yes	N/A
CSV Attendance	Yes	No	Plugin	Yes	No
Docker Deployment	Yes	N/A	Yes	N/A	N/A
Role Dashboards	Yes	Yes	Yes	No	Yes

V. FUTURE SCOPE:

The architecture of the Rural Gyan Platform scales well, as shown in its roadmap which hinges on three key pillars: mobile deployment, production scalability, and AI capabilities. The use of React Native/Expo mobile app will ensure full feature parity with the web version of the platform in light of the fact that over 70% of rural students in India own smartphones. This mobile platform will come with offline-featured quizzes, voice-based interaction with AI chatbots, and compressed media resources for low-end devices and 2G/3G networks. Its over-the-air update strategy through expo allows for continuous improvement within minutes without the need to wait on app stores, leading to an expected increase in the number of rural users by 50%.

Resilience in the production phase will be ensured by leveraging Docker and Docker-compose technologies, thereby allowing for full environment parity in development and production. The stack will support up to 10,000+ users concurrently, with assets on the frontend distributed through Vercel/Netlify CDN and automatically scalable Nodejs instances hosted on Railway or Render services. Zero downtime deployment can be achieved through GitHub actions CI/CD pipelines with an estimated cost of $0.01/user/month.

In addition, future features of the platform will include predictive dropout's analytics, auto-generated curriculum based on AI, multi-lingual support in 10+ Indian languages by leveraging Whisper and NLLB models. AR/VR classrooms through WebXR technology will allow for immersive offline science labs, with TensorFlow.js-based edge inference of AI even in villages with zero connectivity. Analytics generated from data collected through the platform's ELK stack will provide valuable insights for decision-making, while Blockchain-powered certificates will improve graduate employability. All these features put Rural Gyan on track to reach 100 million rural Indian learners by 2027 – aligning perfectly with the UDISE educational guidelines in India.

VI. CONCLUSION:

The exemplary architecture and development of the Rural Gyan Platform clearly reflect the feasibility of implementing a highly optimized full-stack engineering approach explicitly designed to overcome major operational hurdles associated with rural education systems. The platform provides a fully interactive and bilingual environment with a synchronized learning experience based on peer-to-peer WebRTC video communication that does not require extensive server-based stream processing systems. Combining robust localized network protocols with native language translation capabilities instantly addresses the issues related to inadequate bandwidth and major linguistic barriers. The introduction of AI tutoring vectors within the platform creates immediate parity and ensures instant accessibility to education for students from remote areas through an exemplary architecture that balances highly advanced urban technology sectors with vast rural territories within the Indian subcontinent.

VII. REFERENCES:

[1]X. Chen, D. Zou, H. Xie, G. Cheng, And C. Liu, "Two Decades Of Artificial Intelligence In Education," Educational Technology & Society, Vol. 25, No. 1, Pp. 28–47, 2022.
[2]I. Roll And R. Wylie, "Evolution And Revolution In Artificial Intelligence In Education," International Journal Of Artificial Intelligence In Education, Vol. 26, No. 2, Pp. 582–599, 2016.
[3]L. Chen, P. Chen, And Z. Lin, "Artificial Intelligence In Education: A Review," Ieee Access, Vol. 8, Pp. 75264–75278, 2020, Doi: 10.1109/Access.2020.2988510.
[4]B. Garcia, F. Gortazar, L. Lopez-Fernandez, M. Gallego, And M. Paris, "Webrtc Testing: Challenges And Practical Solutions," Ieee Communications Standards Magazine, Vol. 1, No. 2, Pp. 36–42, 2017, Doi: 10.1109/Mcomstd.2017.1700005.
[5]A. K. Das And S. Ghosh, "Real-Time Virtual Classroom Platform: Integrating React, Firebase, Webrtc, And Socket.Io," In Proc. 8th Int. Conf. Electronics, Materials Engineering & Nano-Technology (Iementech), 2025.
[6]B. Sredojev, D. Samardzija, And D. Posarac, "Webrtc Technology Overview And Signaling Solution Design And Implementation," In Proc. 38th Int. Convention On Information And Communication Technology, Electronics And Microelectronics (Mipro), Opatija, Croatia, 2015, Pp. 1006–1009, Doi: 10.1109/Mipro.2015.7160422.
[7]B. Pourebrahimi, K. Bertels, And S. Vassiliadis, "A Survey Of Peer-To-Peer Networks," Computer Engineering Laboratory, Its, Tu Delft, The Netherlands.