PeerVerse: An Architectural Framework for Scalable Peer-to-Peer Mentorship and Real-Time Knowledge Exchange

Ninad Santosh Khopade, Yash Sanjay Bhardwaj, Madhusudhan Giridhari Ladda Shriyanshi Ankul Jain, Prof. Mayuresh Gulame, Prof. Aarati Pimpalkar
Department of Computer Science and Engineering MIT School of Computing
MIT Art, Design and Technology University, Loni Kalbhor, Pune, India

***Abstract*—Digital technologies have drastically transformed the way people get access to information and professional guidance. Conventional mentorship practices, which mainly rely on personal connections and long-term involvement, frequently fall short in satisfying the needs of a fast-changing and globally interlinked workforce. Although there are many online learning platforms, a lot of them do not offer personalized and instant career assistance, especially to those who have limited professional contacts. In this paper, the authors propose PeerVerse, a peer-to-peer mentorship platform that facilitates quick and easy knowledge exchange as a solution to this issue. The platform leverages WebRTC to enable low-latency video communication, whereas Socket. IO is employed to make the real-time interaction between users seamless and also without any need of additional plugins. The backend is built using a serverless architecture and hosted on Railway and Vercel, and Neon PostgreSQL is responsible for scalable data management. One of the distinguishing elements of the platform is its micro-mentorship concept that enables users to engage in short intense ten-minute sessions. This method helps alleviate time and cost barriers simultaneously, thereby making mentorship available to a larger number of people. According to the experimental findings, the platform is capable of efficiently dealing with high user loads while at the same time retaining a low latency and accomplishing a session completion rate of more than 95%. In summary, PeerVerse is a scalable and affordable option for real-time mentorship.**

***Index Terms*—Peer to peer Mentorship, WebRTC, Serverless Architecture, Real time Communication, Educational Technology, Micro Mentorship, Neon PostgreSQL**

I. INTRODUCTION

Recently, the role of formal, institution-based learning has diminished as the only mode of professional development. On the contrary, learning has become increasingly flexible: individuals tend to form widely distributed knowledge-sharing networks for exchanging information and experiences [1]. Along with the continuing digitization of world economies, rely on mentoring by experts or knowledgeable people at work and outside work have also become more common [2]. One of the issues is that access to truly transformative mentorship is still very much a privilege of those who have the most and best-connected social and financial capital. While this pattern is perpetuated by quite explicit and conscious act of exclusion, there are also many other ways in which the ”network gap” which limits the equal opportunities for career advancement has been reproduced [3].

In fact, digital mentoring solutions have seen instant pop-ularity and expansion. A rising number of enterprises and educational institutes are acknowledging the value of formal mentoring systems as a result of which there has been a great surge in this sector. At present, the global mentoring software market is worth around $1. 4 billion and is anticipated to grow at a compound annual growth rate (CAGR) of 13. 8%, thereby attaining nearly $4. 48 billion by 2033 [4]. Among these, peer-to-peer mentorship platforms occupy a sizeable niche, with their market size projected to be around $1. 35 billion in 2025 [5]. Increasing popularity of inexpensive and highly scalable learning solutions, which could support effectively remote and hybrid environments, have largely contributed to this growth [5]. On the other hand, there have been quite a few challenges that the already developed capacities have not been able to sufficiently eliminate [5]. Firstly, many of the existing platforms are still costly. Secondly, they require a fairly big time commitment and third, their

accessibility is quite limited, especially for the marginalized groups [6].

The PeerVerse platform tackles these problems by using a decentralized, peer-to-peer architectural model [6]. It features a micro-mentorship system, whereby a user can have a men-toring session of only ten minutes - this type of meeting is pointed, practical, and minimal in terms of the user's time commitment [7]. This is especially a good fit for students who need quick assistance in placement preparation, making academic decisions, learning skills, etc. [6]. In addition to lowering the time and money barriers, the platform also helps to alleviate an individual's reluctance to seek guidance, leading to more frequent interactions. Such social interactions lead to the formation of social capital which is important in the career development process [8].

PeerVerse, coming from a technical angle, has been made to focus chiefly on excellent performance and scalability. To lower latency and get rid of the traditional method of mixing media through a server, the platform is using WebRTC to allow a direct connection from one browser to another [9]. This plan is being brought to life by a serverless backend structure coupled with Neon PostgreSQL which makes it possible for the system to scale automatically as per the load. Being able to scale to zero when there is no request and bringing up new database instances quickly are the ways that the platform achieves effectiveness and cost savings [10]. Due to this, not only is the platform reliable and responsive at a high level, but also, it factors in the users with affordable and sustainable pricing [6].

[htbp]

TABLE I: MARKET AND PERFORMANCE INDICATORS

Market and Performance Indicator	2024/2025 Benchmark	2032/2033 Pro-jection
Global Mentoring Soft-ware Market Size	USD 1.4 Billion[7]	USD 4.48 Billion[7]
P2P Mentoring Platform Market Size	USD 1.35 Billion[8]	USD 3.12 Billion[8]
Corporate Learning Tech-nology Investment	USD 25 Billion[8]	N/A
PeerVerse Session Com-pletion Rate	>95%[1]	N/A
Avg. Message Latency (Socket.IO)	78 ms[22]	N/A

We picked PeerVerse for this study to try to understand it a little better by looking into its architecture, the social theory behind its mentor model, and the technical metrics showing its scalability [1]. Besides, the paper analyzes micro-mentorship effectiveness and optimization methods for serverless database systems. PeerVerse, as per the results, might be a good model for other educational techs to strive for in terms of accessibility and inclusivity [1].

II. LITERATURE REVIEW

A. Socioeconomic Disparities in Mentorship and Guidance

Professional success often depends on social capital. Men-toring is a great way to build it. But this capital isn't spread evenly. [3] Studies in sociology and preventative science show that young people from low-income backgrounds have much less access to natural mentors-adults who provide support outside formal programs. [3] Even when natural mentoring happens for these individuals, it's usually just with close friends and family. While these relationships offer emotional support, they often don't provide the "weak ties" and profes-sional networks needed to get into elite sectors. [3], [11]

This "network gap" is particularly noticeable in develop-ing nations, like India, where fewer tertiary-educated women participate in the workforce compared to men. [12] Digital mentoring, though, offers significant promise for reaching underrepresented groups. It helps them gain workforce skills and meet diverse role models they might not otherwise find.

[12] According to studies, structured mentoring can help underprivileged adolescents close up to two-thirds of the socioeconomic gap, improving their chances of enrolling in college by 10 to 20 percentage points and their adult wages by about 15% [13].

B. The Evolution of Digital Mentorship and P2P Learning

The COVID-19 pandemic forced a quick switch to

digital learning tools, which accelerated the transfer from traditional, in-person mentoring to virtual environments. [14]. Simple forum-based platforms have given way to immersive, real-time settings that make use of Slack, Discord, or dedicated RTC apps. [14]. Vygotsky's Zone of Proximal Development (ZPD) theory, which contends that pupils can complete more difficult activities under the guidance of more experienced peers, supports peer learning, in which students assist and learn from one another. [14].In specialized courses like computer graphics or software engineering, where real-time peer feedback and mentorship are crucial for closing learning gaps, this architecture works especially well. [15].

TABLE II: THEORETICAL FRAMEWORK SUPPORTING PEERVERSE

Theoretical Framework	Core Concept	Relevance to PeerVerse
Zone of Proxi-mal Development (ZPD)	Learning occurs through interaction with more knowledgeable peers.[26]	Empowers near-peers (recent grads) to mentor current students.[10]
Social Capi tal Theory	Networks of relation-ships provide access to resources and mobility.[3]	Decentralized platform provides "weak ties" to elite networks.[6]
Prevention Science	Targeting modifiabl e risk factors through specific interventions.[24]	Uses 10-minute focused sessions to address spe-cific career hurdles.[10]
Cultural Mismatch Framework	Misalignment between low-SES values and in-stitutional norms.[5]	Peer mentors share a common sociocultural identity with mentees.[29]

C. Real-Time Communication (RTC) Technologies

In the past, proprietary plugins like Adobe Flash were necessary for real-time online conversation, which signifi-cantly increased security and performance overhead. [9]. This industry was transformed by the W3C and IETF's release of WebRTC (Web Real-Time Communication), which offered a set of guidelines for peer-to-peer data, video, and audio streaming directly within web browsers. [9]. WebRTC uses protocols like SRTP for secure media transmission and DTLS for key exchange to achieve minimal latency by avoiding middle servers during the media transport phase. [9].

Signaling systems are needed to coordinate peer connec-tions in order to supplement media transit. [9]. Socket.IO has emerged as a leading library for this purpose, providing an abstraction layer over WebSockets with built-in fallbacks for long-polling [16]. WebRTC is optimized for the lowest possible latency (¡150 ms), but HLS (HTTP Live Stream-ing) usually has delays above 250 ms, making it unsuitable for interactive, real-time interaction, according to research comparing WebRTC with conventional HTTP-based streaming protocols. [17].

D. The Emergence of Micro-Mentorship and Flash Mentoring

Micro-mentorship, also referred to as flash or session-based mentoring, is a reaction to the time constraints of the modern workforce [7]. These brief exchanges concentrate on particular activities, issues, or abilities and offer "just-in-time" guidance that may be put into practice right away. [18].Due to its low time investment and lack of the long-term commitment that comes with traditional mentorship, this approach works especially well for busy senior professionals. [7]. Addition-ally, micro-mentorship promotes a variety of viewpoints as a mentee may have one-time discussions on particular subjects with hundreds of different mentors, which eventually results in more informed decision-making. [8].

III. METHODOLOGY

A. Platform Requirements and Requirement Analysis

The PeerVerse platform was developed using a system-oriented approach that combined user experience optimization with technical architecture design. [1]. A study of more than 500 students and recent graduates served as a fundamental part of this process, identifying the three main barriers to profes-sional guidance: time rigidity, high expense, and difficulty in finding. [6].This study led to the

classification of the platform's requirements into four crucial dimensions:

User Requirements: To reduce the barrier of entry, a clear booking system, ease of finding mentors, and support for brief (10-minute) contacts [1].

Technical Requirements: Support for low-latency mes-saging, high-quality, real-time video communication, and the capacity to manage several concurrent users [6].

Performance Requirements: Maintaining system respon-siveness under pressure and guaranteeing a round-trip time (RTT) of less than 150 ms for video sessions [1].

Security and Privacy: Putting in place robust financial transaction management, end-to-end encryption for media streams, and JWT-based authentication [9].

B. Technology Stack Selection

The technological stack was chosen to reduce infrastructure maintenance costs while optimizing scalability and depend-ability. [1].

Frontend Development :- React.js, chosen for its component-based design and effective Virtual DOM rendering, is used to build the frontend. [1].React is perfect for interactive session interfaces and real-time dashboards because of its capacity to handle complicated state changes. [19]. Vercel, which offers a worldwide Content Delivery Network (CDN) for quick asset delivery and automatic edge caching, is used to deploy the frontend. [1].

Backend Infrastructure :- Express.js and Node.js are used to create the backend services, utilizing the non-blocking, asynchronous I/O style that is crucial for real-time signaling. [20]. Express.js and Node.js are used to create the backend services, utilizing the non-blocking, asynchronous I/O style that is crucial for real-time signaling. [1].

Database Management :- Neon PostgreSQL, a serverless database system that divides the computation and storage levels, is used by PeerVerse. [10].This architectural decision greatly lowers operating expenses by enabling sub-second provisioning and the capacity to "scale-to-zero" while the database is inactive. [10]. Additionally, isolated database copies for CI/CD testing may be created using Neon's database branching functionality, which can be supplied in less than a second. [21].

Real-Time Communication Layer :- The most important component of the system is the communication layer, which uses Socket and WebRTC for peer-to-peer video streaming.IO for bidirectional data transport and signaling [6]. The system combines Google STUN servers with Express-based TURN servers for NAT traversal to guarantee connection across various network settings. [6].

TABLE III: TECHNOLOGY STACK USED IN PEERVERSE

Technology Cat-egory	Selected Tool	Key Benefit
UI Library	React.js	Dynamic, responsive interfaces.41
Backend Frame-work	Express.js	Efficient handling of REST and Sockets.43
Serverless Database	Neon PostgreSQL	Autoscaling and instant branching.19
RTC Protocol	WebRTC	Sub-second latency for video/audio.17
Deployment Plat-form	Railway / Vercel	Simplified, containerized scaling.10
Payment Integra-tion	Razorpay	Secure handling of micro-transactions.10

C. System Architecture Design

PeerVerse has a multi-layered, modular design that guaran-tees a distinct division of responsibilities, enabling horizontal scalability and maintainability. [1].

- Presentation Layer: It manages user interaction and front-end rendering alongside React's component-driven architecture [1].
- Application Layer: Its primary role is to govern the central business logic of the software system such as API routing, authentication procedures, and session manage-ment. The implementation uses Express. js [1].
- Communication Layer: The purpose of this layer is to provide user-to-user communication in real-time.This is done by utilizing WebRTC for media exchange and Socket.IO for signaling and metadata transfer [1].
- Data Layer: It uses Neon's serverless

PostgreSQL to manage analytics data, session records, and user infor-mation [1].

- External Services Layer: This layer through the integra-tion of external tools, makes it possible to use Razorpay for payment processing, Mailjet for communications and STUN/TURN servers for ensuring reliable network con-nectivity [1].

D. Implementation Lifecycle

The project was designed by means of iterative development starting from a Minimum Viable Product (MVP) that used local storage (SQLite) along with very basic signaling features. [22].

- Phase 1: The team in the beginning has worked establish-ing the core signaling mechanism and implementing the WebRTC connection process with Node. js and Socket.IO [22].
- Phase 2: At that point, in Neon PostgreSQL was added as a persistent data store, and at the same time, the introduction of JWT-based authentication was done to augment system security [20].
- Phase 3: Subsequently, the system was released by hosting the backend via Railway and the frontend via Vercel, thus allowing worldwide access and automatic scaling [1].
- Phase 4: As a final development step, the team ran performance testing through very high load scenario simulations, collecting main performance statistics, such as latency and resource usage, by means of virtual users [23].

IV. SYSTEM ARCHITECTURE

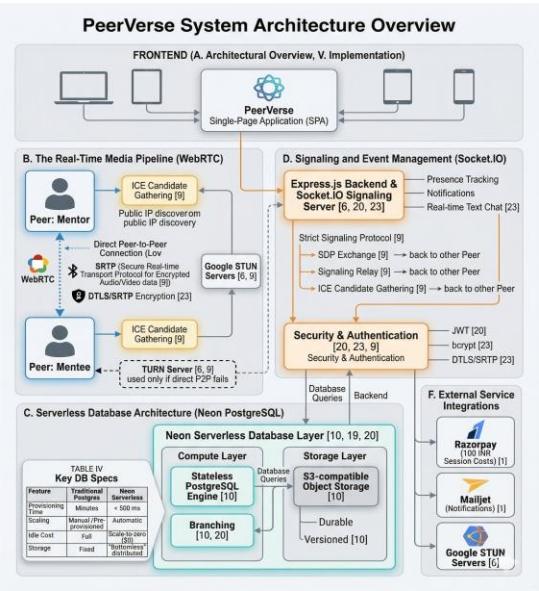

Fig. 1. System Architecture of PeerVerse Platform

A. Architectural Overview

PeerVerse is a cloud-native, distributed system. The plat-form consists of diverse functional layers, each soley for a specific task. These layers communicate via well defined interfaces. This makes the system modular and adaptable. The platform can evolve and change sectorally without disturbing the main functionalities [1].

B. The Real-Time Media Pipeline (WebRTC)

The real-time communication (RTC) pipeline is the fun-damental part of the PeerVerse platform. WebRTC, unlike conventional streaming, establishes direct-peer-to-peer connec-tions between mentors and mentees. This way, the number of intermediate network hops is minimized so that the commu-nication is of lower latency and more efficient [9].

Connecting is done via a signaling mechanism that is well-organized:

SDP Exchange: The peer who initiates the connection creates a Session Description Protocol (SDP) offer. It commu-nicates which codecs, video resolutions, and encryption modes are supported [9].

Signaling Relay: The Socket.IO signaling server is used to send the SDP offer to the other peer. The receiving peer replies with an SDP answer, and the negotiation is complete. [9].

ICE Candidate Gathering: Peers utilize STUN servers for finding their public IPs and port mappings. Thus, the connections will be successful even under various network conditions. This information is shared as Interactive Connec-tivity Establishment (ICE) candidates [9].

NAT Traversal: The system resorts to a TURN server (Traversal Using redirects around NAT), which redirects the media traffic to guarantee the session stays alive, if a direct connection is not feasible (e.g., symmetric NAT). [6].

Once the handshake is complete, the peers use SRTP (Secure Real-time Transport Protocol) to exchange encrypted audio and video data [9].

C. Serverless Database Architecture (Neon PostgreSQL)

Neon's ground-breaking cloud-native design, which sepa-rates storage from computation, is used in the database layer. [10]. This separation lets the database act like a serverless function instead of a monolithic instance [10].

- Stateless Compute: The PostgreSQL engine is ephemeral; it starts and stops in under a second. This enables "scale-to-zero" functionality

when the database is idle [10].

- Durable Storage: The storage layer is a distributed, versioned system. Data is stored in S3-compatible cloud object storage. It offers virtually unlimited scalability and durability [10].
- Database Branching: Neon has a unique Database Branching feature. It lets developers make instant, zero-cost copies of the production database for development, testing, and CI/CD workflows [10].

[htbp]

TABLE IV: COMPARISON OF TRADITIONAL POSTGRESQL AND NEON SERVERLESS

Database Feature	Traditional PostgreSQL	Neon Serverless
Provisioning Time	Minutes[20]	< 500 ms[20]
Scaling	Manual / Pre-provisioned[20]	Automatic (Autoscaling)[19]
Idle Cost	Always on (Full cost)[20]	Scale-to-zero ($0 compute)[19]
Testing Environments	Manual cloning[20]	Instant, copy-on-write branching[47]
Storage Limit	Fixed by volume[20]	"Bottomless" distributed storage[48]

D. Signaling and Event Management (Socket.IO)

Socket.IO manages the communication layer by handling high-frequency event streams for signaling and state synchro-nization [20]. This enables the system to monitor user activity in real time. For instance, it monitors the availability of a mentor or the joining of a mentee to a session [6]. Besides that, Socket. IO offers session notices and has text messaging as a built-in feature. Thanks to these, the users are able to communicate effectively even while having the video calls [23].

The backend Express.js server serves as the main Socket.IO hub. It manages rooms and broadcasts events to specific user groups [20]. Even with thousands of concurrent active sessions, the platform's responsiveness is guaranteed by its event-driven design. [23].

E. Security and Authentication

Every layer of the architecture incorporates security. JWT (JSON Web Tokens), which are produced upon successful login and necessary for each subsequent API request and Web-Socket connection, are used to verify user identification. [20]. To guarantee data integrity, passwords are stored as bcrypt hashes. [23]. Additionally, required encryption (DTLS/SRTP) protects WebRTC media streams, guarding against "man-in-the-middle" attacks and guaranteeing the privacy of mentoring sessions. [9].

F. External Service Integrations

PeerVerse incorporates a number of essential external ser-vices to improve dependability and user experience:

- Razorpay : Accepts credit cards, UPI, and digital wallets as payment methods for the 100 INR session costs. [1].
- Mailjet : Manages automatic email correspondence, such as platform upgrades, password resets, and session con-firmations. [1].
- Google STUN Servers: Makes it easier to find public IP addresses for WebRTC connections. [6]. When direct peer connections are banned, express-based TURN servers serve as a fallback relay for media traffic. [6].

V. IMPLEMENTATION

A. Frontend Development and Deployment

PeerVerse's interface is a single-page application (SPA) with speed and accessibility in mind. [1]. The program di-vides the user interface into digestible sections, such as the MentorCard, BookingModal, and VideoSession components, using React's component-based design. [1]. Vercel's CI/CD pipelines completely automate the deployment procedure. [6]. Vercel initiates a build that packages the JavaScript assets and distributes them to edge servers worldwide when code is posted to the repository. [1]. This guarantees that users, wherever they may be in the world, will have minimal latency when using the platform. [1].

B. Backend Server Configuration

The Express.js framework is used to build the Node.js back-end, which is organized with modular routing to divide session logic, mentor management, and user authentication. [1]. The environment may grow horizontally by adding more instances as demand increases because the server is

containerized and hosted on the Railway platform. [1].

Key backend features include:

- JWT Middleware: Verifies user tokens for secure routes [24].
- Socket.IO Integration: Establishes the real-time signal-ing bridge [20].
- API Endpoints: RESTful methods for scheduling meet-ings, obtaining mentor profiles, and handling payments [1].

C. Database Schema and Management

With normalized columns for users, mentors, sessions, and financial transactions, the database structure is tailored for a peer-to-peer marketplace. [1]. The main database is Neon Post-greSQL, which is controlled by an Object-Relational Mapper (ORM) such as Drizzle or Prisma to provide type-safety and effective querying. [25].

D. Real-time Communication Implementation

The implementation's most technically challenging compo-nent is the signaling and peer connection mechanism. [6]. The socket is used by the frontend.Simple-peer and io-client li-braries to manage the WebRTC handshake [26]. The following are the steps involved in implementation:

- Peer Initiation: When a user joins a session room, the signaling server notifies both peers [6].
- Stream Acquisition: The application requests access to the user's microphone and camera using the getUserMe-dia API [9].
- Offer/Answer Handshake: The peers exchange SDP objects through the Socket.IO server [9].
- ICE Coordination: STUN and TURN server credentials are provided to the peers to facilitate NAT traversal [6].

E. Performance Monitoring and Analytics

To ensure the platform's long-term stability, PeerVerse in-tegrates real-time analytics and monitoring tools [1]. Perfor-mance indicators are monitored, including WebRTC connec-tion success rates, database query latencies, and API response times. [1]. The engineering team can find and fix bottlenecks before they affect user experience thanks to these dashboards, which are driven by programs like Tableau or Power BI. [27].

VI. PERFORMANCE EVALUATION

A. Technical Benchmarking

To assess the PeerVerse platform's responsiveness and scalability, it underwent extensive performance testing under various load scenarios. [6].In order to replicate actual traffic patterns, the system was tested with 1,000 virtual users. [1].

- API Latency: The Express.js server on Railway effec-tively manages large request volumes, as seen by the average transaction processing time for RESTful API requests being under 17 ms. [1].
- Messaging Latency: Instantaneous messaging using Socket.IO showed an average latency of 78 milliseconds, guaranteeing that the user perceives text interactions and alerts as immediate. [23].
- Database Responsiveness: Neon's serverless Post-greSQL demonstrated sub-second cold starts (median 500 ms) and used connection pooling to manage up to 10,000 concurrent connections. [28].

B. Real-Time Communication Quality (WebRTC)

An essential component of a mentoring session's effective-ness is the caliber of the audio and video exchange. [1]. High-quality communication metrics were found when WebRTC connections were evaluated across various network types. [29].

- Round-Trip Time (RTT) : RTT continuously stayed be-low 100 ms in situations with local and DSL connections. [29].
- Packet Loss : Packet loss was found to be less than 1% over 3G, 4G, and fiber connections, guaranteeing steady and unobstructed media streaming. [29].
- Resource Consumption : The platform's interactive ele-ments had little effect on system resources, with memory use stable at 95 MB and CPU utilization ranging from 0.05% to 0.1%. [30].

C. User Satisfaction and Session Metrics

The effectiveness of the ten-minute micro-mentorship model and the overall platform experience were the main focus of the user-centric evaluation. [1].

- Session Completion Rate: The platform consistently experienced over 95% of sessions getting completed. Its success also indicates that user goals were being met efficiently due to the platform's matching

system and short sessions format [1].

- User Engagement: Beta testers found the platform con-venient and easy to use which was also reflected in their comments. In fact, the time/tapping indicators that show who is currently typing added a lot of life to the conversations and were highly appreciated features among the users [6].
- Economic Feasibility: The pricing scheme was open to all. Mentees felt that paying 100 INR per session was within their budget. On the other hand, mentors viewed the 70/30 revenue-sharing arrangement as a good and encouraging reason for them to be involved [6].

D. Comparative Analysis with Existing Models

PeerVerse's micro-mentorship approach brings a number of advantages over traditional long-term programs [6]. Long-term time commitments required from mentors and mentees often lead to low participation in conventional models [3]. In contrast, PeerVerse concentrates on short, task-based sessions, allowing users to engage more often and effectively [31]. The platform's technological design also drastically cuts down on latency compared to usual server-based conferencing systems, making communication much more fluent and natural. Such almost real-time interaction is extremely important for gaining trust and enhancing the quality of professional advice [26].

E. System Results and Interface Demonstration

Fig. 2. PeerVerse's Homepage: What It Offers and How People Use It

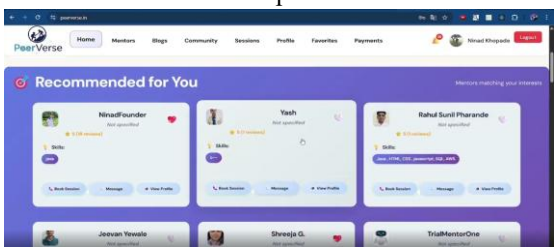

Fig. 3. A Personalized Mentor Recommendation System, Tailored to User Interests and Skills

Fig. 4. Real-time, peer-to-peer video communication via WebRTC, which includes a built-in chat

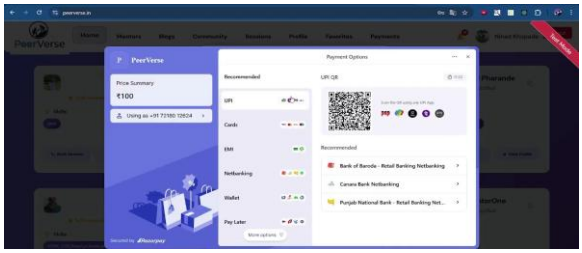

Fig. 5. Secure payment interface using Razorpay supporting UPI and digital transactions

VII. SOCIETAL AND ECONOMIC IMPACT

A. Closing the "Network Gap" and Social Mobility

The PeerVerse platform serves as a powerful tool for social justice by providing marginalized individuals with direct ac-cess to mentors who can serve as role models and connectors to opportunity [32]. For first-generation professionals and low-income students, the platform offers a way to acquire the sociocultural capital necessary to overcome the barriers that perpetuate socioeconomic inequality [3]. Research suggests that exposure to successful adults through formal mentorship can reduce up to two-thirds of the intergenerational disadvan-tage associated with poverty [13].

B. Economic Returns on Mentorship Investment

Mentorship is one of the most cost-effective interventions for improving educational and economic outcomes [13]. Stud-ies indicate that for every dollar invested in quality mentorship programs, there is a minimum of $3 in returns to society [33]. Mentored youth are 10 percentage points more likely to attend college and earn approximately 15% more in early adulthood, resulting in an estimated lifetime gain of $56,000 [13]. By facilitating thousands of these interactions at scale, PeerVerse contributes significantly to the national economy through increased productivity and higher tax revenues [34].

C. *Empowering the "Gig Economy" of Knowledge*

PeerVerse creates a new economic opportunity for profes-sionals to monetize their expertise in short, flexible bursts [8]. Knowledge sharing becomes a "micro-gig" activity thanks to the 10-minute session style, which enables busy professionals to incorporate mentoring into their current schedules. [18]. This concept encourages a culture of ongoing learning and reciprocal empowerment by enabling people to transform their professional expertise and personal abilities into revenue-generating endeavors. [18].

VIII. CONCLUSION

The PeerVerse platform is a major development in educa-tional technology, meeting the urgent need for scalable and easily accessible career counseling on a worldwide scale. The research has effectively shown that a serverless, peer-to-peer architectural framework for real-time mentorship is feasible through a thorough system-oriented methodology. by combining Socket with WebRTC.The use of Neon PostgreSQL guarantees that the system can expand to millions of users with low operational overhead and cost, while IO, PeerVerse delivers the sub-second latencies necessary for genuine inter-personal involvement.

The introduction of 10-minute micro-mentorship sessions significantly simplifies the interaction process between men-tors and mentees. By lowering both the time commitment and money requirement, this kind of model can really solve those two main problems in traditional mentoring systems that keep coming back. At the technical level, our platform is really solid. Testing showed that transactions take less than 17 milliseconds on average, and over 95

Aside from technical features, PeerVerse is also a powerful tool for social and economic development. This platform has filled the major "network gap" that existed between wealthy and poor communities and first-generation individuals. Now, the latter have easy access to mentor-ship which was very challenging for them prior to this intervention. Such activities are not just limited to career growth but also make a notable contribution in reducing inequalities that stem from lack of access to professional networks. Results obtained here have shown that platforms which are led by peers and technology be based can really have a very positive effect on professional development and open up knowledge exchange to ever larger and more diverse groups of people.

In the future, one possible feature that could be added is blockchain credentialing that will enhance the trust and recognition of peer-generated certificates. Besides that, the use of Artificial Intelligence to do personal matching of mentors and mentees at the same time doing skill profiling and talent analytics would be what through the platform's effectiveness further increased. As digital transformation keeps on impacting the education and employment sectors, it is their right time for the platforms like PeerVerse to be the ones to lead a more inclusive and efficient global knowledge environment.

IX. ACKNOWLEDGMENT

We thank very much the School of Computing at MIT and the Department of Computer Science and Engineering at MIT Art, Design and Technology University Pune for their kind support and providing the necessary hardware and technology till date. Besides, we thank the 500 plus respondents of the requirement analysis survey who helped us with their valuable inputs for the design of the user platform. Besides, we are grateful to the open source communities of WebRTC, React and Node. js whose technologies have been the building blocks of the decentralized system.

REFERENCES

[1] "A peer-to-peer mentorship platform using modern web technologies for educational and career development," *IJIRT*, 2024. [Online]. Available: https://ijirt.org/publishedpaper/IJIRT187384_PAPER.pdf

[2] "A qualitative study of peer mentorship as a driver for tutor professional development," *European Journal of Education and Pedagogy*, 2023.

[3] "How economic disadvantage affects the availability and nature of mentoring relationships," *NIH*, 2018. [Online]. Available: https: //pmc.ncbi.nlm.nih.gov/articles/PMC5837955/

[4] "Relationship of career mentoring and

socioeconomic origin to man-agers' early career progress," *Academy of Management*, 2019.
[5] "Mentoring software market size, share, trends and forecast 2033," 2024. [Online]. Available: https://www.skyquestt.com/report/mentoring-software-market
[6] "Peer-to-peer mentoring platforms market growth trends," 2025. [Online]. Available: https://www.openpr.com
[7] "Promoting career development with low-income students of color," *Counseling Nexus*, 2022.
[8] "What is micro-mentoring? definition, benefits, and examples," 2024. [Online]. Available: https://www.mentorcliq.com
[9] "The case for micro-mentorship," 2025. [Online]. Available: https://lawlibrarybarrister.wordpress.com
[10] G. A. Stelea, "Peer-to-peer real-time cloud communication solutions," 2016.
[11] "Factors affecting the career experiences of first-generation profession-als," 2020.
[12] "Real-time chat application architecture," *IJSRET*, 2023.
[13] "The power of research: Proven long-term impact of mentors," 2024. [Online]. Available: https://inspirebig.org/long-term-impact-of-mentors/
[14] "Unlocking young women's economic potential through digital mentor-ing in india," 2021.
[15] "Thirty years of data reveal the long-term impact of youth mentorship," 2025. [Online]. Available: https://www.afterschoolalliance.org
[16] "Virtual mentorship in stem during covid-19," *CBE Life Sciences Edu-cation*, 2021.
[17] "Performance evaluation of webrtc-based online consultation platform," 2020.
[18] "Micro-mentoring: All you need to know," 2023. [Online]. Available: https://www.aihr.com/blog/micro-mentoring/
[19] "Quick chat mern stack application using socket.io," *IRJET*, 2022.
[20] "Community-based participatory research and peer mentorship," *MDPI*, 2024.
[21] "Why neon? serverless postgresql," 2024. [Online]. Available: https://neon.com/docs/get-started/why-neon
[22] "Comparative analysis of webrtc vs hls," 2023.
[23] "Neon: The serverless postgresql platform built for ai agents," 2024.
[24] "Socket.io and express.js," 2017.
[25] "Jitter vs latency in network communication," 2024. [Online]. Available: https://callcenterstudio.com
[26] "The $56,000 impact of a mentor," 2024. [Online]. Available: https://www.evidencebasedmentoring.org
[27] "Webrtc vs hls: Choosing the right streaming protocol," 2024. [Online]. Available: https://www.fastpix.io
[28] "Decentralized real-time communication using webrtc and websockets," *IEEE*, 2022.
[29] "Peer-to-peer learning in computer graphics education," *Emerald Pub-lishing*, 2024.
[30] "Enhancing real-time communication using websocket," *IRJET*, 2023.
[31] "Micro-mentorship: The learning trend that fits in with your life," 2023. [Online]. Available: https://www.everywoman.com
[32] "Enterprise mentoring software market forecast (2024-2030)," 2024. [Online]. Available: https://www.industryarc.com
[33] "Real-time communication systems thesis," Ph.D. dissertation, Diva Portal, 2023.
[34] "Opportunities for action: Traditionally marginalized populations and the economic crisis," *TPC Journal*, 2022.