Grenze International Journal of Engineering and Technology, June Issue

Empowering Educators and Learners: A Video Conferencing Application for Online Learning

Megha Jain[1],Ayush Singhal[2], Aishwary Bhadauriya[3] and Harshit Pratap Singh[4]
[1-4]JSS Academy of Technical Education, Noida, India
meghajain12@gmail.com , ayushsinghal2912@gmail.com ,aishwarybhadauriya09@gmail.com , singhharshitpratap@gmail.com

***Abstract*— The current market lacks a video conferencing software that seamlessly integrates an IDE, resulting in inconvenience and time consumption for users who have to switch between different applications. Our research aims to provide a solution to this problem by developing an all-in-one video calling app that includes an IDE, real-time whiteboard, and file sharing options. This research paper describes a video calling app that includes an integrated development environment (IDE) as part of its software. The program is intended for usage in meetings and interviews, allowing users to code without switching to another software. It also has a live whiteboard and file sharing capabilities, making it a flexible tool for collaborative work. The application has the power to completely change how online interviews are conducted and how distant teams and educational institutions communicate. We aspire to have a positive influence on how people work and learn in the quickly evolving world of today by offering a free and open platform for remote coding and collaboration.**

***Keywords*—videoconferencing, integrated IDE, Real-time file share , Web Real-Time Communication**

I. INTRODUCTION

In today's remote work and learning environments, video calling apps have become a necessity for effective collaboration. However, many of these apps require users to switch to other software to code or collaborate on documents, which can be inconvenient and time-consuming. To address this problem, we developed a video calling app that contains all the basic features of existing video calling apps, such as screen sharing and audio and video communication, as well as additional feature that integrates an IDE into its software, allowing users to code directly within the app. The app also includes a real-time file sharing and whiteboard option, making it a versatile tool for collaborative work. In this paper, we describe the development and testing of this app, and discuss its potential benefits for remote teams and virtual learning environments. Our goal is to provide a more streamlined and efficient solution for coding and collaboration within a video calling platform(Judge and Neustaedter, 2010).

This app has many potential applications, particularly in the fields of education and tech. In schools and colleges, the app can be used for online classes and exams, and for conducting interviews and tests. In tech companies, the app can be used for remote collaboration, code reviews, and meetings. Our app provides a streamlined and efficient solution for remote collaboration, learning, and conducting interviews online. The COVID-19 pandemic has

Grenze ID: 01.GIJET.9.2.376

highlighted the importance of remote work and learning, and we believe that our app can make this transition easier and more effective. By providing a single platform for video conferencing, coding, and collaboration, our app eliminates the need to switch between multiple applications, thus saving time and effort for users(Edigo, 1988).

Communication and collaboration have become essential in the digital age, especially in the education sector. The need for remote learning and virtual classrooms has led to the development of numerous communication tools. EduCare is one such comprehensive solution that offers a range of features to facilitate remote learning and collaboration(Singh and Jain, 2015).

The idea for our research project came during the COVID-19 pandemic, when many organizations and educational institutions were forced to shift to remote work and learning environments. We observed that existing video conferencing apps lacked an integrated development environment (IDE), which is essential for remote coding and collaboration. This led us to develop an all-in-one video calling app that seamlessly integrates an IDE, real-time whiteboard, and file sharing options(Drexhage, Schmidt and Ehmke, 2016).

Our motivation for this project is to address a critical gap in the market for an all-in-one video calling app with an integrated IDE. We believe that our app has the potential to revolutionize the way remote teams and educational institutions collaborate and conduct online interviews. By providing a free and accessible platform for remote coding and collaboration, we hope to make a positive impact on the way people work and learn in today's rapidly changing world(Freeman, 1998).

II. Literature Review

The purpose of this literature review is to compare the effectiveness of remote education via video conferencing versus traditional education in classrooms. Various methods of research investigation like problem, logical, comparative analysis, synthesis, and formalization research investigation methodologies, as well as a purposefully developed author's way of assessing the success of distant education and traditional education.

(Vucic, Skorin-Kapov and Suznjevic, 2016) suggested by researchers that the importance of education through the use of video conferencing. In this rapidly changing environment, it is necessary to improve and provide education to as many students as possible, and remote education or teaching through the use of video conferencing here plays a very important role because institutions or teachers will have no limitations, and students will be able to reach and learn from teachers all over the world.

(Kasetwar *et al.*, 2022)The COVID-19 pandemic has had a tremendous influence on education, with schools and universities all across the world shifting to online or remote learning to combat the virus's spread. Platforms for video conferencing have played an important part in this transformation, allowing students and professors to continue their education from a distance has been presented by the researcher.

(Ng *et al.*, 2008) suggested by researchers that Videoconferencing systems are a prominent tool for online or remote learning in higher education. They enable students and teachers to connect and cooperate in real time from many places by combining video, audio, and, in some cases, text or other information. There are several videoconferencing systems available, ranging from free web-based programmes such as Google Meet or Zoom to more complex, dedicated systems commonly used in educational institutions. These systems often include a number of capabilities that can aid in online learning and collaboration, such as screen sharing, recording, and annotation.

(Edan, Al-Sherbaz and Turner, 2019) suggested by researchers that Video conferencing technology can be a valuable tool for teacher training in a connected classroom. It allows educators to participate in professional development opportunities remotely, making it easier for them to access high-quality training and resources.

(Fai Ng *et al.*, 2014) suggested by researchers that WebRTC (Web Real-Time Communication) is a powerful open-source technology that enables real-time communication through web browsers. One of the most popular applications of WebRTC is video conferencing, which allows people to communicate with each other through video and audio. A WebRTC based video conferencing system with screen sharing is a particularly useful tool for remote teams, businesses, and educational institutions, as it allows participants to collaborate effectively regardless of their physical location

(Augestad and Lindsetmo, 2009) suggested by researchers that Bandwidth constraints can have an impact on the quality of video and audio streams being sent. If there is insufficient available bandwidth, the video may be blocky or pixelated, and the audio may be choppy or have a high latency. This can make it difficult for participants to communicate effectively, lowering their QoE. Similarly, the video resolution size might have an effect on QoE. Higher quality video needs more bandwidth to send, and if there is insufficient available

bandwidth, the video may be downscaled to a lower resolution to compensate. This might lead to a lower-quality video, which can be aggravating for participants.
(Kydd and Ferry, 1994)There has been significant growth in the usage of video-conferencing (VC) among medical workers, especially surgeons who have embraced the technology. Total of 517 articles were found, with 51 relevant manuscripts containing the key words. Telementoring surgical operations, as well as trauma and emergency medicine, are common uses for VC among surgeons has been presented by the researcher.
(Asan, Smith and Montague, 2014) suggested by researchers that significance of education in young people with disability has been searched, the pros and cons of teaching these students via the use of video conferencing has been researched. The flexi e-learning system is designed with the view in mind to provide a common platform where the abled and disabled both can study and learn together.
(Nakatsuhara *et al.*, 2017)conferencing software is becoming increasingly significant in the workplace, education, and social contact. This necessitates the accessibility and usefulness of these instruments. For example, are the primary features completely accessible to all users? Researchers investigated the accessibility and usability of screen readers and keyboards by individuals with visual impairment in this study.
(Judge and Neustaedter, 2010)A P2P-MCU (Peer-to-Peer - Multipoint Control Unit) approach to a multi-party video conference with WebRTC (Web Real-Time Communication) involves using a combination of both P2P (peer-to-peer) and MCU (multipoint control unit) architectures to facilitate the video conference. In a P2P architecture, each participant in the conference connects directly to every other participant in the conference, and all participants can send and receive media streams directly from one another has been presented by the researcher.
(Vucic, Skorin-Kapov and Suznjevic, 2016) suggested by researchers that WebRTC (Web Real-Time Communication) is a technology that enables real-time communication in web browsers and other applications via peer-to-peer connections. It supports audio and video streaming as well as data sharing without the use of plug-ins or third-party applications. A signaling mechanism is a procedure that allows two or more devices to initiate and sustain an internet communication session. A signaling system is used in WebRTC to establish a connection between two or more devices and communicate session information, such as the kind of media to be transferred, the codecs to be used, and the addresses and ports to be utilized for the media stream.

III. System Software Requirements And System Architecture

In this paper, we aimed to create a video calling application with an integrated IDE, which required us to consider the technical feasibility of our plan. To create the front-end of our web application, we decided to use React.js, a popular JavaScript library known for its flexibility and performance. For the back-end, we chose Node.js, another widely-used JavaScript runtime environment that allows for easy server-side scripting. To facilitate low-latency and bidirectional communication between users, we utilized Socket.io, a popular library that provides real-time event-based communication. We also used WebRTC, a standard that enables real-time communication in web browsers without the need for additional plugins or software. With WebRTC, we were able to incorporate audio and video streams into our application, and we used RTC Data Channel interface to enable bidirectional data transfers between users(Fai Ng *et al.*, 2014).
To create the integrated development environment (IDE) within our web application, we utilized NPM Editor Packages, which allowed us to add various code editing features such as syntax highlighting, autocomplete, and debugging tools. We also used Docker to provide a compiler environment for executing code in different languages within our web application(Jain, Pandey and Kewal Krishna Sharma, 2022a).
To establish a peer-to-peer connection between users, we used STUN and TURN servers, which enabled us to discover the IP addresses of users and facilitate communication between them. We also created a website for end-users to interact with our application.
Finally, we considered the system requirements for our application, including a working webcam and microphone, a minimum of 2 GB of RAM, a quad-core processor, and a minimum of 1 Mbps network bandwidth. By carefully considering these technical requirements and implementing the appropriate protocols, we were able to create a fully functional video calling application with an integrated IDE.
The EduCare system architecture is based on a client-server model, where the clients access the EduCare platform using a web browser. The platform is built using React.js for the frontend and Node.js for the backend. The backend uses a signaling server and sessions database for establishing and maintaining real-time communication between clients. The clients use WebRTC for real-time communication, including audio and video streams, and the RTC Data Channel interface is used for bidirectional data transfer between clients(jain, Kaswan and Pandey, 2021).

In addition to real-time communication, the EduCare platform includes various widgets such as code editor, drawing, screen sharing, file sharing, rich group/private messaging, and slide sharing. These widgets are implemented as components of the frontend using NPM editor packages and Docker to provide a compiler environment for executing codes of different languages(Jain, Pandey and Kewal Krishna Sharma, 2022b).
The session layer components include a policy controller, calendar, notifications, authentication, address book, presence, and session analytics. These components ensure that the communication and collaboration on the EduCare platform are secure, organized, and efficient.
Overall, the EduCare system architecture is designed to provide an end-to-end efficient and powerful communication solution for users, with a focus on seamless integration and user experience.
The system is divided into three layers - signaling, widget, and session. The signaling layer comprises a signaling server and a session database that manages the peer-to-peer offer, answer, and SDP transactions. The widget layer includes components such as direct calls, code, draw, context awareness, screen sharing, file sharing, group messaging, auto-notes, real-time analytics, and a SlideShare widget. Finally, the session layer has session analytics, policy controller, calendar, notifications, authentications, presence, and a host of other components(Jain, Pandey and Krishna Kewal Sharma, 2022).
EduCare's features are designed to facilitate communication and collaboration between teachers and students, enabling them to share ideas, thoughts, and feedback. The system supports direct calls, which allows users to call anyone on their network, making it easy to connect with peers or teachers for assistance. Code, draw, and context-awareness widgets help users collaborate in real-time, while screen and file sharing features allow for effective sharing of information.
EduCare's policy controller allows administrators to set policies for sessions, services, and users, providing an additional layer of security. The system also supports authentication, which ensures that only authorized users can join a session. Presence and session analytics help administrators monitor and manage the platform effectively.

IV. METHODOLOGY

EduCare's architecture is designed to be modular and scalable, making it easy to integrate with other systems. The system can be accessed through any desktop or mobile browser that supports WebRTC, and the API and SDK are developer-friendly.To develop EduCare, a set of components required to build an efficient and powerful communication solution, the following methodology will be adopted:
A. Requirement Gathering: The first step in the methodology is to gather requirements for the EduCare system. This involves understanding the needs of the users, determining the features required, and defining the system architecture.
B. Design: Once the requirements are gathered, the design phase begins. This includes the creation of high-level and low-level design documents, which outline the system's architecture and components.
C. Implementation: The implementation phase involves the actual coding of the system. This involves writing the code for the different modules and integrating them into a single system.
D. Testing: Testing is an essential aspect of software development, and it is no different for EduCare. The testing phase involves testing the system to ensure that it meets the specified requirements and that there are no bugs or errors.
E. Deployment: After the system has been tested and verified to be free of errors, it is deployed in a production environment. This involves setting up servers, installing software, and configuring the system.
F. Maintenance: Once the system is deployed, it will require maintenance to ensure that it continues to function optimally. This includes periodic updates, bug fixes, and enhancements to the system.
G. The methodology outlined above will be used to develop EduCare. Each phase will be carefully planned and executed to ensure that the final product meets the needs of the users and is of high quality

V. RESULT

A link is provided after the user selects "Create Room," and that link may be shared with attendees so they can quickly join the same room or meeting where the host is already present. Each user in a room is given or allocated a unique ID via a random id generator, which the backend uses to verify the specific room of which a user posted a link. In this manner, the meeting's security and privacy are not compromised, and the host may make sure that only real persons are allowed in the space. An additional layer of security is enabled by this verification and permission.

Once connected and in the room, the user can see and speak with the person on the other end, share files in real time, and chat with them. Video and audio feeds are also shared, and features like background blur and a whiteboard allow for simultaneous teaching and writing in the meeting. Additionally, users can use the chat box function to chat and share messages.

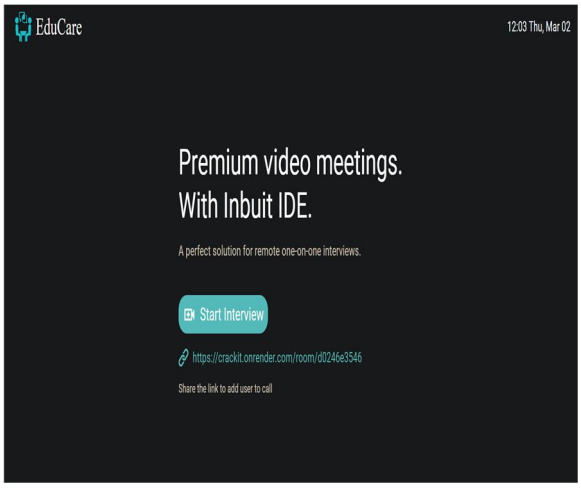

Fig. 1: Landing page

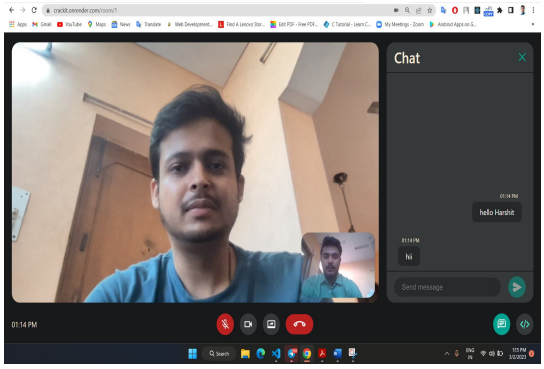

Fig 2: Meeting Room

The host or user can share their screen and code in many languages using an IDE (Integrated Development Environment), which also has features like autocomplete and formatting options that make it easier for programmers to create and compile code. C++ code was written and compiled, and input and output screens were used. The output screen displayed the results, and the input screen took the input.

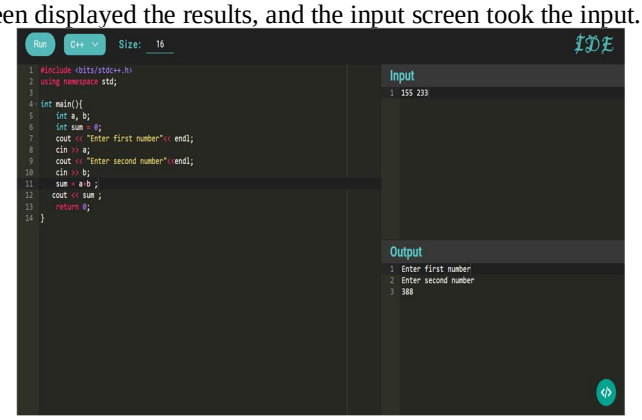

Fig 3: IDE to write code.

VI. Conclusion

In conclusion, the goal of our research study was to create an all-in-one video calling software with an integrated development environment (IDE) for remote coding and team collaboration. Since users had to hop between several applications, we saw that the existing video conferencing apps were missing this crucial feature, which caused inconvenience and wasted time of users. A complete platform for online interviews, learning, and remote collaboration is offered by our app. Our tool removes the need to switch between various applications, saving users time and effort by combining an IDE, a real-time whiteboard, and file sharing features effortlessly.

We think that our application has the power to completely change how online interviews are conducted and how distant teams and educational institutions communicate. We aspire to have a positive influence on how people work and learn in the quickly evolving world of today by offering a free and open platform for remote coding and collaboration.

Our study demonstrates that our product fills a critical market demand for an all-in-one video calling app with an integrated IDE. We hope that our work stimulates more investigation and advancement in this field, ultimately resulting in more simplified and effective options for online learning and cooperation.

References

[1] Asan, O., Smith, P. D. and Montague, E. (2014) 'More screen time, less face time - Implications for EHR design', Journal of Evaluation in Clinical Practice, 20(6), pp. 896–901. doi: 10.1111/jep.12182.

[2] Augestad, K. M. and Lindsetmo, R. O. (2009) 'Overcoming distance: Video-conferencing as a clinical and educational tool among surgeons', World Journal of Surgery, 33(7), pp. 1356–1365. doi: 10.1007/s00268-009-0036-0.

[3] Drexhage, J., Schmidt, D. L. T. and Ehmke, T. (2016) 'The Connected Classroom: Using Video Conferencing Technology to Enhance Teacher Training', Reflecting Education, 10(1), pp. 70–88.

[4] Edan, N. M., Al-Sherbaz, A. and Turner, S. (2019) 'WebNSM: A novel scalable WebRTC signalling mechanism for one-to-many Bi-directional video conferencing', Advances in Intelligent Systems and Computing, 857(November), pp. 558–568. doi: 10.1007/978-3-030-01177-2_40.

[5] Edigo, C. (1988) 'Videoconferencing as a Technology to Support A Review . of its Failure Group Work ':, Computer Supported Cooperative Work (CSCW), pp. 13–24.

[6] Fai Ng, K. et al. (2014) 'A P2P-MCU Approach to Multi-Party Video Conference with WebRTC', International Journal of Future Computer and Communication, 3(5), pp. 319–324. doi: 10.7763/ijfcc.2014.v3.319.

[7] Freeman, M. (1998) 'Video conferencing: A solution to the multi-campus large classes problem?', British Journal of Educational Technology, 29(3), pp. 197–210. doi: 10.1111/1467-8535.00064.

[8] jain, M., Kaswan, S. and Pandey, D. (2021) 'A Blockchain based Fund Management Scheme for Financial Transactions in NGOs', Recent Patents on Engineering, 16(2), pp. 1–14. doi: 10.2174/1872212115666210615155447.

[9] Jain, M., Pandey, D. and Sharma, Kewal Krishna (2022a) 'A blockchain approach on security of health records for children suffering from dyslexia during pandemic COVID-19', Artificial Intelligence, Machine Learning, and Mental Health in Pandemics: A Computational Approach, pp. 343–363. doi: 10.1016/B978-0-323-91196-2.00004-1.

[10] Jain, M., Pandey, D. and Sharma, Krishna Kewal (2022) 'A Granular Access-Based Blockchain System to Prevent Fraudulent Activities in Medical Health Records', Lecture Notes on Data Engineering and Communications Technologies, 106, pp. 635–645. doi: 10.1007/978-981-16-8403-6_58.

[11] Jain, M., Pandey, D. and Sharma, Kewal Krishna (2022b) 'A Granular Approach to Secure the Privacy of electronic Health Records Through Blockchain Technology', International Journal of Distributed Systems and Technologies, 13(8). doi: 10.4018/IJDST.307899.

[12] Judge, T. K. and Neustaedter, C. (2010) 'Sharing conversation and sharing life: Video conferencing in the home', Conference on Human Factors in Computing Systems - Proceedings, 2, pp. 655–658. doi: 10.1145/1753326.1753422.

[13] Kasetwar, A. et al. (2022) 'A WebRTC Based Video Conferencing System with Screen Sharing', International Journal for Research in Applied Science and Engineering Technology, 10(5), pp. 5061–5066. doi: 10.22214/ijraset.2022.43595.

[14] Kydd, C. T. and Ferry, D. L. (1994) 'Managerial use of video conferencing', Information and Management, 27(6), pp. 369–375. doi: 10.1016/0378-7206(94)90017-5.

[15] Nakatsuhara, F. et al. (2017) 'Exploring the Use of Video-Conferencing Technology in the Assessment of Spoken Language: A Mixed-Methods Study', Language Assessment Quarterly, 14(1), pp. 1–18. doi: 10.1080/15434303.2016.1263637.

[16] Ng, C. K. et al. (2008) 'Flexi e-learning system: Disabled friendly education system', European Journal of Social Sciences, 7(2), pp. 120–127.

[17] Singh, V. and Jain, M. (2015) 'Analysis of trust dynamics in cyclic mobile: Ad hoc networks', 2015 1st International Conference on Futuristic Trends in Computational Analysis and Knowledge Management, ABLAZE 2015, pp. 400–406. doi: 10.1109/ABLAZE.2015.7155029.

[18] Vucic, D., Skorin-Kapov, L. and Suznjevic, M. (2016) 'The impact of bandwidth limitations and video resolution size on QoE for WebRTC-based mobile multi-party video conferencing', pp. 59–63. doi: 10.21437/pqs.2016-13.