iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VI **Month of publication:** June 2022

DOI: https://doi.org/10.22214/ijraset.2022.43930

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue VI June 2022- Available at www.ijraset.com

Video Calling with Build-In Compiler

Nikhil C K[1], Enugu Veneeth Reddy[2], Jonnalagadda Rudra Naga Sai[3], Kaliga Vinay Kumar[4], Sowmya Varshini Avula[5] , V Maheshwar Reddy[6]

[1, 2, 3, 4, 5]*CSE, Department Of Computer Science & Engineering, ACE Engineering College, Ghatkesar, Hyderabad, Telangana, India.*

[6] *Assistant Professor ,Department of Computer Science and Engineering, ACE Engineering College, Ghatkesar, Hyderabad, Telangana, India*

***Abstract:** In today's Internet world, many applications that were previously created on a desktop computer are now being created on the web. Many applications can be accessed anytime and anywhere easily using the Internet. Due to the pandemic, the job recruitment process has moved online via Zoom, Google Meet, and other video calling apps. But during every IT interview, the person who is giving the interview faces major problems such as one must share their screen via a video calling app and open a compiler to run the program given by the interviewer. If the student doesn't have a compiler installed on his system, then they need to open a compiler in a web browser to compile the code.*

The purpose of this research is to design and develop a real-time code editor application using Web socket technology to help users collaborate during tech interviews. There is a feature in this application that allows video calling with a built-in compiler. Video calling with a built-in compiler is a web application that provides a workspace to write, perform, display the results of the code through the terminal, and collaborate with other users in real-time. It provides workspaces for creating, executing, and building source code, as well as for real-time collaboration, chat, and building terminals. This application supports C, C++, Java, and Python programming languages.

I. INTRODUCTION

Software engineering has experienced an improvement in technological development where the design of software has moved from the desktop to the web. Due to the digitalization of the world things that were once performed manually are completely transformed into a digital environment. All the activities that were once done offline are now available through the Internet. Because of the Covid 19 outbreak, all activities were conducted online. One such activity that was severely affected was the recruitment process. The traditional method of recruiting face-to-face could not be applied in the present situation, so digital recruitment was done. The interviews were usually conducted online through applications like Google Meet, Zoom and many more. But those applications had a major flaw: they thought of how the interview is done in general.

Conducting tech interviews through these video calling applications does not seem ideal as they miss key features that are needed for conducting interviews smoothly is presence of is the compiler or the IDE (Integrated Development Environment). Furthermore, most compilers were desktop-based and therefore still had significant disadvantages, such as taking a long time to load and having to install plug-ins if multiple languages are being used. This problem is a huge waste of time during the interview process for both the host and participant and hence therefore there is a need for a pair programming video calling application that can be used during the interview which allows both host and participant to have a free flow interview where both can edit, delete, write and compile the code in any language they want in a hassle-free manner.

II. LITERATURE SURVEY

Programmers while working on development projects in a team if has the access to the project can change, add code in the same project file. For the process to be smooth without any ambiguity such as same code repeating twice, synchronization is required. This problem can be solved by Integrated real-time collaboration in a one environment. IDE (Integrated Development Environment) provides collaborative platform for programmers who work in a team for real-time text editing, run and write code, compile, and execute the code etc. The advantage of real-time text editing is that it enables programmers or users to work together, and changes can be seen to all other users who are working or has access to that same document. There are other few applications that has enabled this feature such as Google Docs. Not only this feature makes up for a very good and effective programming but awesome collaboration for users.

©IJRASET: All Rights are Reserved

Project or software development demands coordination and communication among programmers; So, tools for collaboration are necessary for efficient and effective project. Collaborative programming increases the efficiency and quality of a project or piece of software. Real-time collaborative programming allows programmers simultaneously to work on the same file or code. This feature automatically merges the code which is being written by the programmers without programmer executing commands such as update, commit. Two or more programmers at a time can access and edit the same source code. A programmer can communicate with other users by joining and exiting a real-time session. Using the join button, a programmer can join the session and on clicking exit button, user can leave the session.

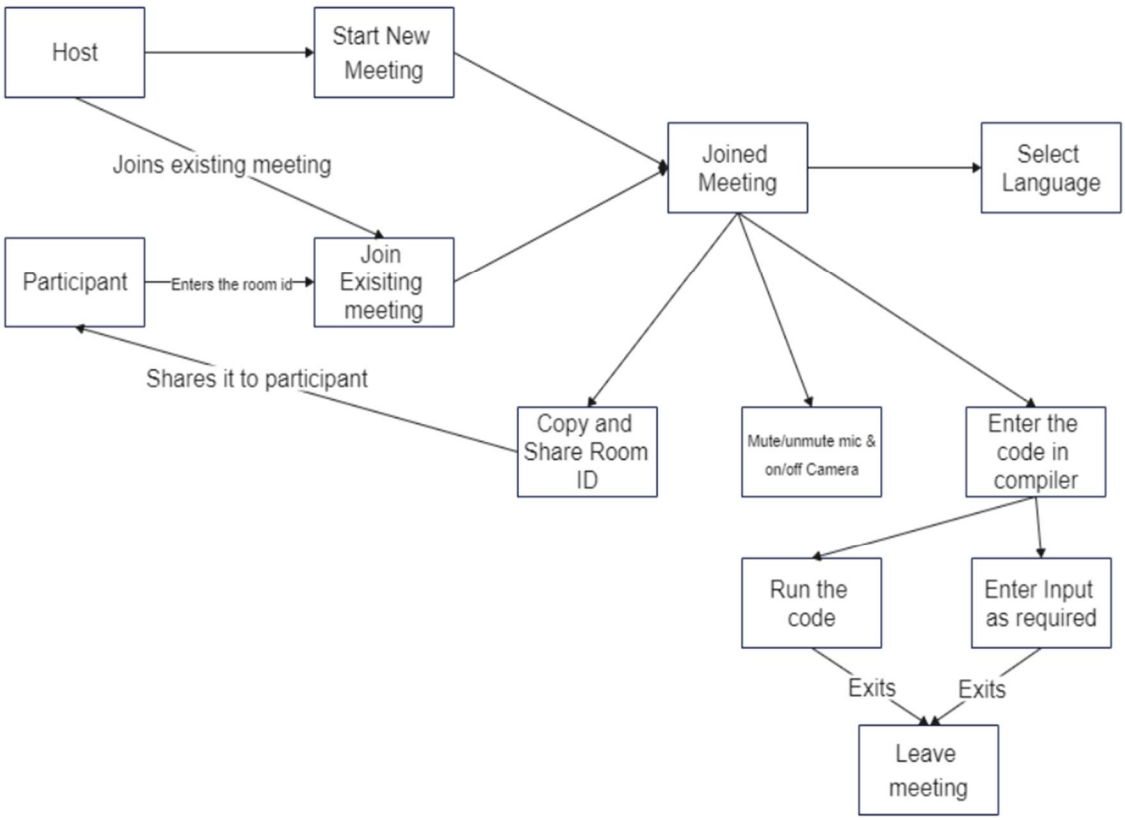
III. EXISTING SYSTEM

Collaboration can also be accomplished through a wide variety of web-enabled systems. It is possible to edit text in real-time using EtherPad, for instance. Ace and CodeMirror are web-based text editing components designed to integrate with an IDE or application. But these are only real-time text editing web-based systems without video and audio features. There are many video calling applications which are used for interviews, such as Microsoft Teams, Google Meet, Cisco Webex meetings, and zoom. These provide high end communication features that are extensively used and needed for interview purposes, but in case of Tech interviews for programming questions, these applications do not provide a compiler to execute the program.

IV. PROPOSED SYSTEM

The proposed system is a web-based application. Host (Interviewer) first enters the website application and hence starts a new meeting, and a new room is created with a randomly generated 128-bit value used to uniquely identify an object. This bit value is then sent to the participant (interviewee) who can enter the unique bit value to join the room. Proposed System contains integration of video calling and compiler. The proposed system allows host and participant to both voice and video call rights when they join the web-based system. Whenever the host asks a coding question, then the participant can directly write the code in the build-in compiler, selecting a language of their choice, compile the code, and obtain the result.

V. SYSTEM ARCHITECTURE

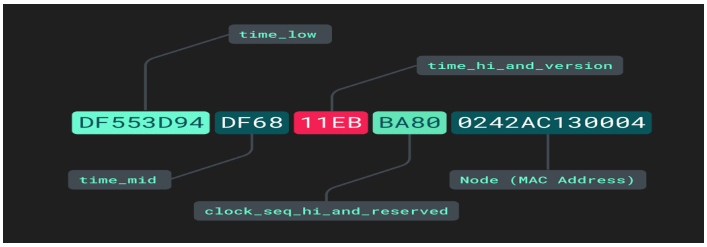


VI. IMPLEMENTATION

A. Dependencies Used

1) Uuid Version 8.3.0

Any collection of data within a computing system can be identified uniquely by a Universally Unique Identifier (UUID), which is an alphanumeric 36-character designation.UUIDs are a widely used method for granting persistent and unique identities due to their extremely low likelihood of duplication to about any form of data that may be imagined. UUID is a one-of-a-kind 128-alpha-numerical identifier. This is the label that is used as the room id.



2) Socket io Version 4.5.0

Socket.IO is a library that enables bidirectional, event-based and low-latency communication between a client and a server.

It is based on the WebSocket protocol and includes features such as HTTP long-polling fallback and automatic reconnection. Socket.IO is a real-time web application library based on an event-driven JavaScript language. It allows web clients and servers to communicate in real time. It is divided into two parts: a browser-based client library and a Node.js server library. The APIs are substantially same in both components.

3) Peer JS version 1.3.1

Peer JS is a web development dependency that covers the WebRTC implementation and hence allows peer-to-peer connection of API's. P2P data or media connection stream is possible that even allows peer remote connection.

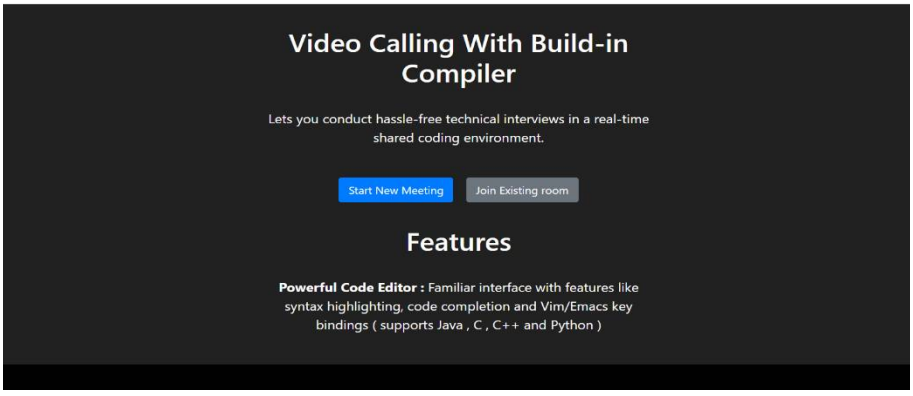
4) Cors Version 2.8.5

Through Cross-Origin Resource Sharing (CORS), we can remove security constraints placed on APIs. Specifically, it is done by ignoring the Use-Control-Allow-Origin header, which tells the API which sources are allowed to access it. Essentially, it controls what domains can access your resources and limits cross-origin HTTP requests to other servers.

B. Overall Interface

1) Login Interface

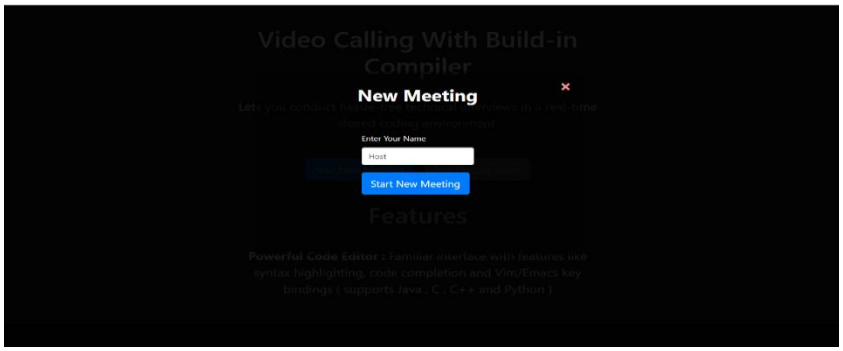
Both the host and participant are introduced to the home page when they enter into the website.



2) Start New Meeting

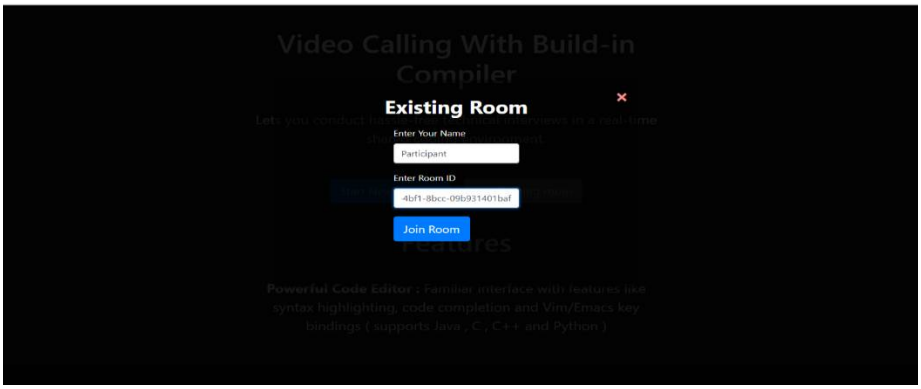
The host can start a new meeting. The host can be the interviewer in case of interview scheduling. After clicking on Start New Meeting, the host will need to enter his name. This name will be visible throughout the session until the user is present in the session.

When a host starts a new meeting, a 128 unique room id is created which needs to be copied and send it to the participant.



3) Join Existing Meeting

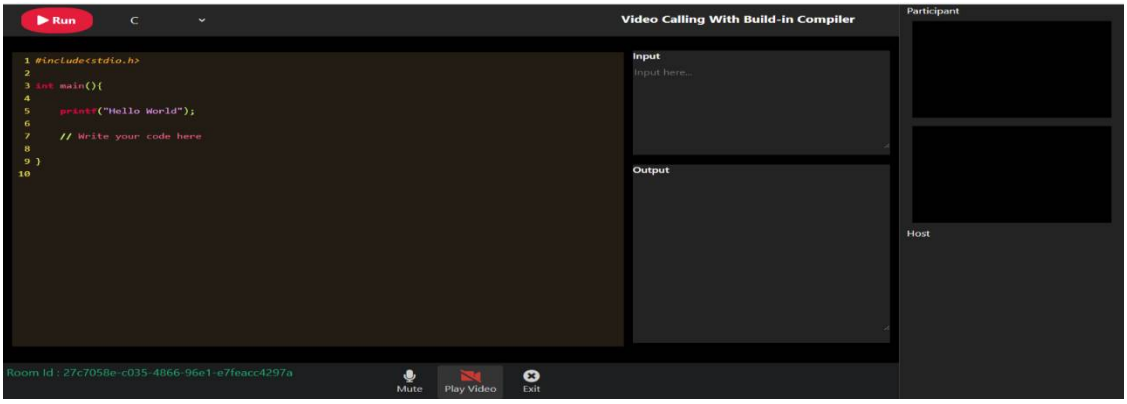
The participant can join an existing meeting. The participant (interviewee in case of interview) needs to by click on Join Existing Meeting and hence enter his name and enter the room id that has been shared by the host. A participant can login only if the room id matches with the existing room id created.



4) Logged in Interface

After both the host and participant login, they have the similar interface that consists of the compiler, input region and output region.Both the host and the participant can change the language as per their needs between C, C++, Java, and Python and run the code by clicking on Run button.

On the bottom of the screens, we have the room id present along with three buttons to on/off camera, mute/unmute mic and leave the room.The right side of the screen contains the video of both host and participant that is captured through camera.



VII. WORKING

Both the host and participant are connected to each other through an end-to-end connection; hence each of them can communicate with one another without any issue via the internet. In this case, socket io dependence may result from the end-to-end connection.

```
io.on("connection", (socket) => {
  console.log('connection')
  socket.on("join-room", (roomId, userId) => {
    socket.join(roomId);
    socket.to(roomId).emit("user-connected", userId);


    socket.on("code", (message) => {
      //send
      socket.to(roomId).emit("code", message);
    });

    socket.on("inpmsg", (message) => {
      socket.to(roomId).emit("inpmsg", message);
    });
    socket.on("outmsg", (message) => {

      socket.to(roomId).emit("outmsg", message);
    });

    socket.on("disconnect", () => {
      socket.to(roomId).emit("user-disconnected", userId);
    });
```

Inside the compiler, real time typing, and competition can be done from both ends i.e., all the input that is typed by the participant is shown to the host in real time and vice versa. This is possible with the usage of Judge0 compiler.
Connection between the host and participant compiler is possible with the use of Judge0 API key which allows seamless, fast, and robust connection and transfer of data with one another.

```
const API_KEY = "91382a6df9msh35c7f3d0d2833b6p195bd1jsn904e1954e5f6";

function check(token) {
  $("#output").val($("#output").val() + "\n⬇ Checking submission status...");
  $.ajax({
    url: `https://judge0-ce.p.rapidapi.com/submissions/${token}?base64_encoded=true`,
    type: "GET",
    headers: {
      "x-rapidapi-host": "judge0-ce.p.rapidapi.com",
      "x-rapidapi-key": API_KEY,
    },
    success: function (data, textStatus, jqXHR) {
      if ([1, 2].includes(data["status"]["id"])) {
        console.log("Hello i am here u are where ----------");
        $("#output").val($("#output").val() + "\n⏳ Status: " + data["status"]["description"]);
        setTimeout(function () {
          check(token);
        }, 1000);
      } else {
        var output = [decode(data["compile_output"]), decode(data["stdout"])].join("\n").trim();
        $("#output").val(`${data["status"]["id"] != "3" ? "🔴" : "🟢"} ${data["status"]["description"]}\n${output}`);
        $("#run-btn").prop("disabled", false);
      }
    },
    error: errorHandler,
  });

  setTimeout(function () {
    const text = output.value;
    socket.emit("outmsg", text);
  }, 2500);
}
```

VIII. RESULT

Several mock and open tech-based interviews were tested between a one-to-one interview consisting of a single host and a single participant, and a one-to-many interview consisting of many hosts and single participants. Most of the users were satisfied with the interface, performance, and overall working of the web-based compiler. Many suggested improvements in the UI are currently being worked upon. The judge0 compiler produces accurate output like any currently available compiler on the market. The audio and video calling features also produced high-quality output without any interference with a stable internet connection between host and participant. With the input from the users, we have also started working on building new features into the web-based application that shall be rolled in future updates.

IX. CONCLUSION

Traditional methods might be effective in the short term, but over time they will need to update and move towards providing better services using technology. As a result of our project, we observed that most people were satisfied with having resources in one place. The emphasis of this paper is to allow easy, hassle-free interviews in the domain of tech interviews for both the users i.e., host and participant. The web-based application that we've built has solved most of the problems the existing system is facing and in the upcoming years we will try to improve on other factors if required.

©IJRASET: All Rights are Reserved

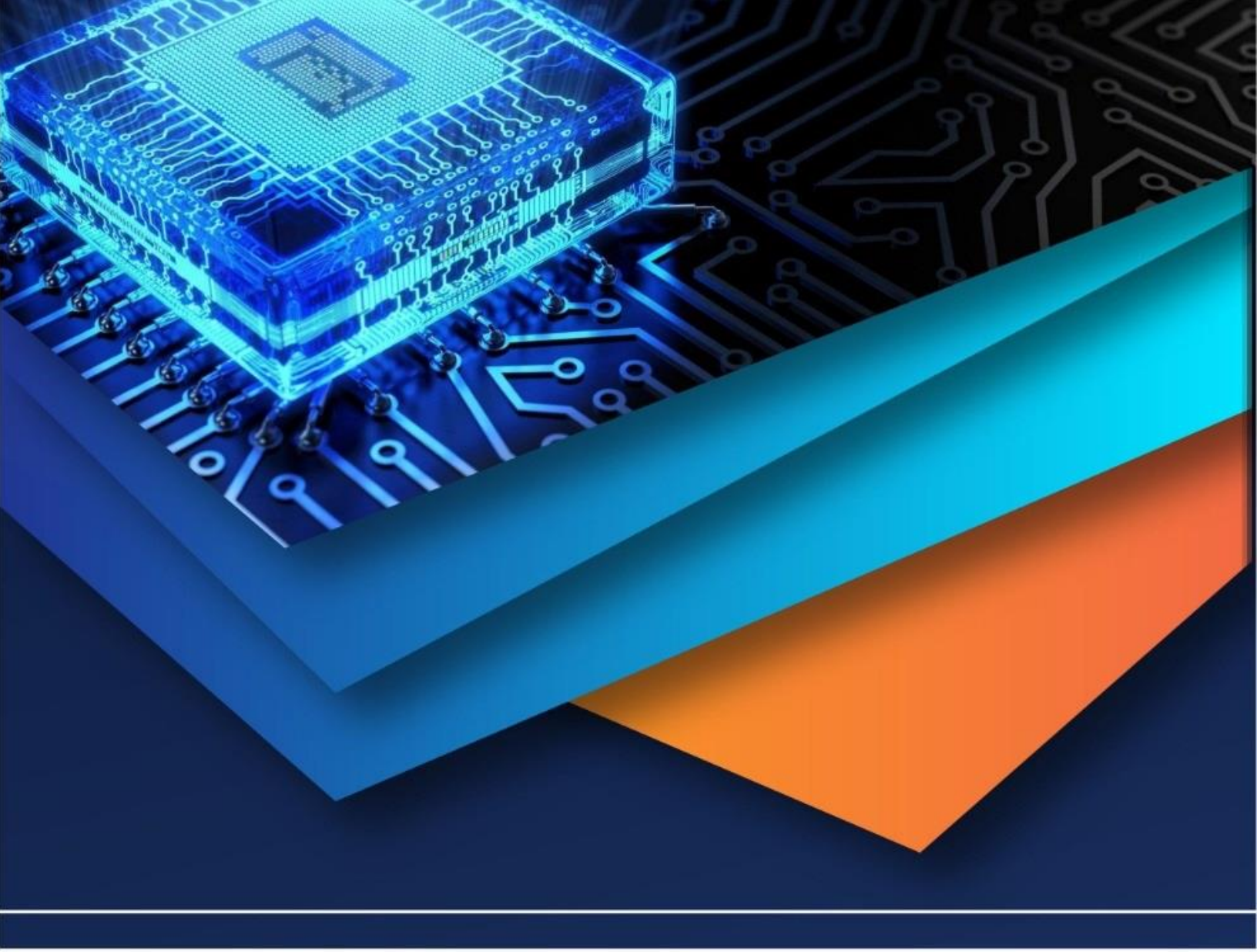
X. ACKNOWLEDGEMENT

We are grateful to our guides Associate Prof. Mr. V. Maheshwar Reddy and Associate Prof. Mrs. Soppari Kavitha for their continuous support and guidance. Through their guidance, we were able to successfully complete our project. Our sincere thanks go to Dr. M. V. VIJAYA SARADHI, Head of the Department of Computer Science and Engineering at Ace Engineering College, for his support and time.

REFERENCES

[1] Rohit Rai in Socket. IO Real-Time Web Application Development (E book) pages displayed by permission of Packt Publishing Ltd. Copyright.
[2] P. Leach, M. Mealling, R. Salz in A Universally Unique IDentifier (UUID) URN Namespace dated July 2005Copyright (C) The Internet Society (2005).
[3] Mr. Ravinder Singh and Soumya Awasthi in Updated Comparative Analysis on Video Conferencing Platforms- Zoom,Google Meet, Microsoft Teams, WebEx Teams, and GoToMeetings a EasyChair Preprint- No 4026.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)