Journal of Science, Computing and Engineering Research (JSCER)
Volume-8, Issue-4, April 2025.
DOI: *https://doi.org/10.46379/jscer.2025.080403*
ISSN: 2708-1079

ONLINE CODING ENVIRONMENT FOR INTERVIEWS USING NETWORK API

[1]Mohit Sharma, [2]Tarun Kataria, [3]Nitin

[1] Assistant Professor, Department of CSE, Modern Institute of Technology and Research Centre, Alwar, Rajasthan, India
[2,3] UG Student, Department of CSE, Modern Institute of Technology and Research Centre, Alwar, Rajasthan, India

Article Information

Received : 20 April 2025
Revised : 21 April 2025
Accepted : 23 April 2025
Published : 25 April 2025

Corresponding Author:
Tarun Kataria

Abstract— This In the era of remote work and virtual hiring, technical interviews require platforms that combine real-time coding collaboration with face-to-face communication. This paper introduces an Online Coding Environment designed to address this gap. The platform enables multiple users — including interviewers and candidates — to collaboratively write and execute code in real-time, while communicating through integrated video calls. It supports multiple programming languages (JavaScript, Java, Python, and C) and uses modern web technologies like React.js, Node.js, Socket.IO, WebRTC, and MongoDB. The system enhances the assessment process by unifying communication and problem-solving in one seamless environment.

Citation: Mohit Sharma, Tarun Kataria, Nitin, "ONLINE CODING ENVIRONMENT FOR INTERVIEWS USING NETWORK API", Journal of Science, Computing and Engineering Research, 8(04), April 2025.

I. INTRODUCTION

The demand for remote hiring solutions has significantly increased, especially in the software industry. Traditional interviews, where candidates solve problems on paper or whiteboards, do not reflect modern collaboration practices and become difficult to manage in virtual setups. Hence, organizations are shifting toward online platforms where candidates can code live, receive feedback, and demonstrate their thought process.

However, most platforms either focus solely on code execution or on video communication — very few offer a complete, integrated solution. This project aims to fill this gap by providing an environment where users can write, share, execute code, and conduct technical interviews with video call support, all within the same browser window. The system uses Network APIs for real-time interaction, multi-language code execution, and secure video communication, which collectively create a productive and interviewer-friendly environment.

This project — Online Coding Environment for Interviews Using Network API — is an innovative web-based solution designed to address this evolving need. It is a real-time collaborative platform that allows multiple users, including interviewers and candidates, to write, edit, and execute code together during technical assessments. Beyond just collaborative coding, this system also integrates video call functionality, enabling face-to-face interaction between interviewers and candidates within the same virtual workspace. This paper proposes an Online Coding Environment that addresses these issues through integrated tools and seamless communication options. The platform offers code compilation for JavaScript, Python, Java, and C, enhancing candidate convenience and interviewer flexibility.

II. PROBLEM STATEMENT

The main problem faced by modern interviewers and candidates is the lack of a unified environment that supports both code collaboration and direct communication. Often, candidates and interviewers must rely on separate platforms for video conferencing and coding, which increases confusion, disrupts flow, and limits interaction. A single platform that offers these combined capabilities can simplify the interview process, reduce time wasted on technical setup, and improve the evaluation of candidates' problem-solving abilities.

The lack of dedicated platforms that enable collaborative, real-time coding interviews with integrated code execution and communication tools hampers both the interviewer's assessment and the candidate's presentation. This project aims to build an intuitive environment that solves these constraints while supporting multiple programming languages.

III. PROPOSED METHOD

The proposed system uses the MERN stack as its backbone (MongoDB, Express.js, React.js, Node.js) and

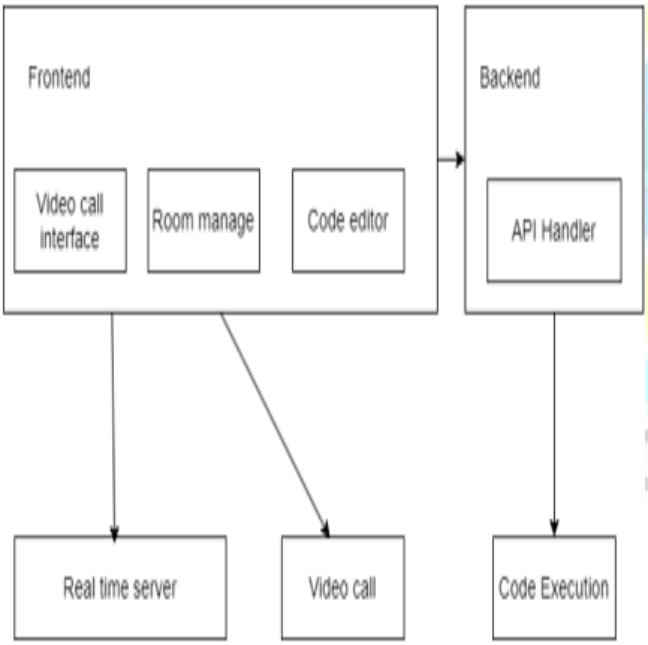
enhances it with **Socket.IO** for real-time code collaboration and **WebRTC** for secure video calling.

The proposed system is built using MERN Stack (MongoDB, Express.js, React.js, Node.js) integrated with WebSocket and WebRTC for real-time collaboration and video communication. This allows for a seamless interview experience with features like live code execution, chat, and whiteboard.

Key Features:

- Real-time code collaboration with multiple users.
- Multi-language support: JavaScript, Java, Python, and C.
- Peer-to-peer video calling integrated in the same interface.
- Dynamic room creation using unique room IDs.

IV. COMPONENT DIAGRAM

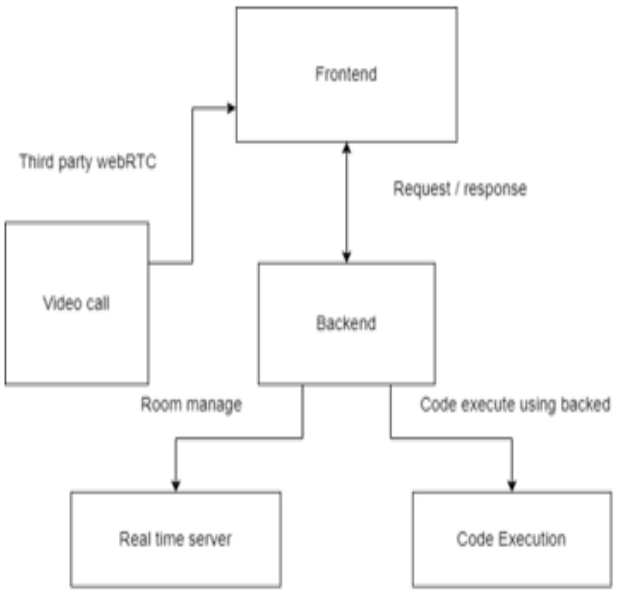


V. IMPLEMENTATION

The platform was developed using a combination of modern tools:

- Frontend: React.js + Tailwind CSS.
- Backend: Node.js, Express.js.
- Real-time Collaboration: Socket.IO.
- Video Calls: WebRTC.

VI. BLOCK DIAGRAM:



Sample Code Snippet:

Sample code how to work language for compilation on server side -

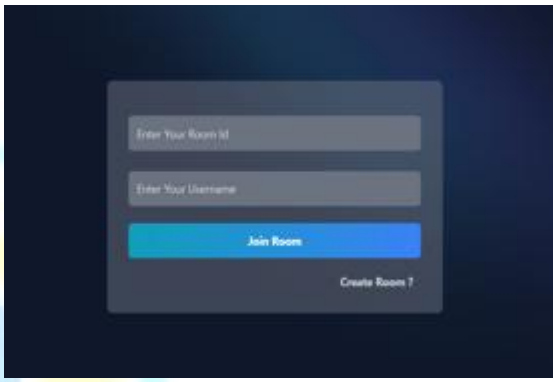
```
async function executeCode(code, language) {
 if (language === 'javascript') {
   try {
    const timeoutMilliseconds = 5000;
    const sandbox = { console }; // Include the console object for logging
    vm.createContext(sandbox);

    // Capture console output
    const executionTimeout = setTimeout(() => {
     // If execution exceeds the timeout, throw an error
     console.log("hello error")
    }, timeoutMilliseconds);

    let output = '';
    sandbox.console.log = (data) => {
     output += data + '\n';
    };
    // Execute the JavaScript code in the sandbox
    vm.runInContext(code, sandbox);
    return output;
  } catch (error) {
    console.error('Error while executing code:', error);
    return `Error: ${error.message}`;
  }
 }
 try {
  const response = await axios.post(
```

```
    'https://judge0-ce.p.rapidapi.com/submissions?base64_encoded=false&wait=true',
    {
      language_id: getLanguageId(language),
      source_code: code,
    },
    {
      headers: {
        'content-type': 'application/json',
        'X-RapidAPI-Key': RAPIDAPI_KEY,
        'X-RapidAPI-Host': RAPIDAPI_HOST,
      },
    }
  );
  return response.data.stdout || response.data.stderr || 'No output';
 } catch (error) {
  console.error('API Error:', error);
  return `Error: ${error.message}`;
 }
}
function getLanguageId(language) {
 switch (language) {
  case 'javascript':
    return;
  case 'python':
    return 71; // Python 3
  case 'java':
    return 62; // Java
  case 'c_cpp':
    return 54; // C++
  default:
    return null;
 }
}

app.post('/execute-code', async (req, res) => {
 const { code, language } = req.body;

 if (!code || !language) {
  return res.status(400).json({ error: 'Code and language are required.' });
 }

 try {
  const output = await executeCode(code, language);
  res.json({ output });
 } catch (error) {
  res.status(500).json({ error: error.message });
 }
});
```

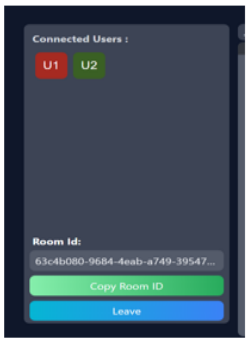
VII. SCREENSHOT:

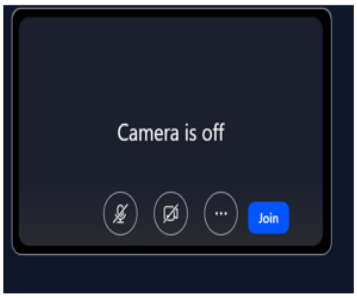
VIII. RESULTS

Room joining or creation



User or interviewer lists and video call





Coding editor with output

IX. CONCLUSION

The project titled Online Coding Environment for Interviews Using Network API has successfully demonstrated a real-time, collaborative platform that bridges the gap between traditional in-person interviews and the modern needs of remote technical assessments.

By integrating technologies like Socket.IO for real-time collaboration, WebRTC for video calling, and supporting code execution across multiple languages such as JavaScript, Java, Python, and C, the platform provides a complete and interactive experience for both interviewers and candidates. The user-friendly room-based system eliminates the need for complex account setups or authentication, allowing users to focus directly on problem-solving and communication.

Through testing, the platform proved its capability to handle collaborative interviews effectively, with smooth code synchronization and clear peer-to-peer video communication. Its simplicity and efficiency make it a promising solution not only for interviews but also for remote learning, hackathons, and collaborative software development.

REFERENCES

[1]. Alan Green et al. (2021) explored the implementation of real-time communication tools in their paper “Building Real-Time Communication Tools for Technical Interviews”
[2]. Sarah White and her team, in “Interviewing in the Cloud: A Study on Remote Coding Assessments” (2022)
[3]. conference paper “Leveraging APIs for Scalable Online Coding Platforms” by Michael Brown et al. (2023)
[4]. Wang et al. (2021) titled “Real-Time Collaboration and Interview Performance: The New Metrics in Software Hiring”
[5]. "Collaborative Programming in the Cloud" by Smith et al., 2021