Code Doodle: An AI-Powered Collaborative Coding & Communication Platform

Harsh Gupta[1], Aditya Kumar[2], Harjender Singh[3]

Abstract:

The problem our team finds out that during COVID-19, due to lockdown, no one is allowed to do work from the office due as a result, professional life shifted to WFH, work from home, and due to this, professionals, especially the developer software engineers, face lack of communication while building the product because engineers. They are not able to collaborate, and many engineers handling the module have their own tasks to integrate when all work gets done.

The second issue my Team Member analyses that there is no one single platform which is providing the code-review assistant, face-to-face meeting support, and real-time code collaboration and file saving directory and developer need to use separate web apps for different needs like google meet for face-to-face meeting and codeshare web app for real-time code share editor which reveal the issue.

So, Code Doodle Web application is planned to be developed by our team to resolve those mentioned issues. It provides Real-time code sharing platform, Face-to-face meeting, File saving feature, Detecting who deleted the code to maintain the discipline, and Code- Reviewer.

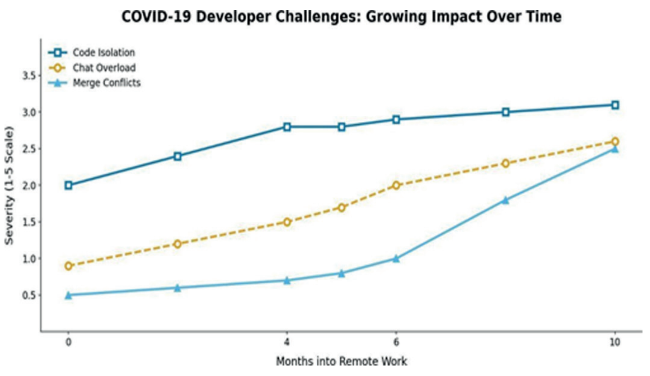
The technologies that we used are socket.io for maintaining real-time changes in code, WebRTC for conducting face-to-face meeting, Gemini API for building code-reviewer and chatbot, Mongo Database for storing and retrieving the user login and registered details, node.js for creating the Backend Server, and React.js for the frontend part.

The key feature is that we can provide our buggy code to code reviewer it tell the bugs inside it and gives suggestion to improve the code, 24X7 available Meeting is supported to conduct face-to-face communication in the common room setup by socket to discuss about the project, Provides also the functionality that who so ever in the code editor room try to delete the code will be informed to all rest of the developer and provide the dialog box to ask whether to save it or not in recycle bin. The significance of it is that all the problems is now solved by providing the Single Platform which does not exist yet, and in future our team is trying to add some new features like Compiler to run the code and dashboard where they can check their saved files.

Keywords: Collaborative Coding ,Real-Time Communication ,Remote Work ,COVID-19 ,Merge Conflicts ,Socket.io ,MongoDB , AI-Powered Code Review ,Code Consistency ,WebRTC.

Introduction & Objectives

During COVID-19, due to lockdown, no one is allowed to do work from the office as a result, professional life shifted to WFH, work from home, and due to this, professionals, especially the developer software engineers, face lack of communication while building the product because engineers. They are not able to collaborate, and many engineers handling the module have their own tasks to integrate when all work gets done due to which, during the integration of those small modules of code, there is a higher chance of having the Merge code conflict and code consistency is disturbed and face huge bugs during development and are not able to meet the project deadlines.



There is no single platform that is providing code review assistant, face-to-face meeting support, and real-time code collaboration and file saving directory, and developers need to use separate web apps for different needs, like Google Meet for face-to-face meetings and Codeshare web app for a real-time code share editor, which reveals the issue. This

1. *Research Scholar. MSI, harsh01721202023@msijanakpuri.com*
2. *Research Scholar, MSI, aditya03021202023@msijanakpuri.com*
3. *Assistant Professor, MSI, harjendersingh@msijanakpuri.com*

forces the developer to keep jumping between different apps all day, like their code editor, a chat app to communicate, and a video call. This made it impossible for them to focus on writing the code. All these problems messed up the code, and merge conflict arises, thus they couldn't finish their projects on time.

We develop an all-in-one platform for remote collaboration. With Code Doodle's real-time synchronized code editor, merge conflicts don't happen at all because everyone in the room is able to view the same file changes at once and even save the file before someone tries to delete it. The significance of it is that all the problems is now solved by providing the Single Platform which does not exist yet.

Objectives of the Project

a. Primary Objectives

- **Code Collaborative editor:** It provides the panel which is to every developer who entered the Room and it is secured by the Unique Key generated. Every change appears in real time to each developer who joined the same so as to maintain the Code Consistency and eliminating merge conflicts.
- **Familiarize with & integrate AI-based code review using Gemini API:** Panel accepting buggy code and find the bug & provide suggestion to improve it.
- **One to One Video Calling using WebRTC:** Implement the One-to-One video calling with hang-up, mute/ unmute, camera-on/off feature in order to make the environment more professional.
- **Recycle Bin & Reliable :** Providing the reliable panel by informing the developer in that room if someone try to remove the existing code and saving it in recycle bin.
- **Built-in chat system for communication:** Chat support to ask for doubts about anything, especially related to code improvement.
- Secure access through JWT authentication. Providing a secured Registration, Login & Logout feature with sending mail and verifying the user before allowing him to use the website tools.

b. Secondary Objectives

- **Provide a user-friendly, modern interface:** Making the website's UI easy to understand and provide instructions to use the various tools with ease for users.
- **Deliver smooth animations using GSAP and Framer Motion:** Integrate Smooth Transition and flow for it to be more user-responsive and user-friendly using GSAP and Framer- motion.
- **High performance and scalability:** High-performance through the usage of NoSQL databases, that are faster than conventional SQL for storing and retrieving user information. This provides horizontal scalability by storing information in JSON format.
- **Allow downloading and saving of code files:** Providing the Downloading and saving functionality on our workspace and recycle bin before someone tries to delete it in the code editor room.

Scope & Theoretical Background Scope of Project

The project scope involves:

1. Features Included

- **Real-time collaborative code editing:** Provide the panel where every change in the code is shown in real time to every developer who joined the same room in order to maintain Code Consistency.
- **AI-based code review module:** It provides an AI-based code review feature, through which one can review code, find the mistake, and provide suggestions to improve it.
- **Implemented Chat system & video calling:** Chat system and Video calling is separately provided where a developer can conduct face-to-face meeting which makes the remote work more professional.
- **Secure Authentication with email verification:** In this project, we have used token- based authentication to make sure that the registered user is authorized to use those tools which we are providing in Code doodle.
- **Interactive UI with dark/light mode:** Light and Dark mode is provided in order to make the UI more user-friendly.
- **Integrating Gemini AI in chatbot and code analysis:** Code Reviewer is additional feature that our team is providing to make all the problems that a developer faced in on place. It takes the buggy code and provides a suggestion to improve it.

2.. Features not Included-Future Scope

- **Multi-user video conferencing:** Since WebRTC only support peer-to-peer connection, so in the future Code doodle's team will try to provide support for multi- user video conferencing.
- **Screen sharing and whiteboarding:** We are planning to add one more feature in our project, where the user can draw something, explain the logic also shown on the other developer in the room connected for better coding logics.
- **Integrated compiler for multiple languages:** Currently, there is no compiler provided where developer can run their code and test the same. But in future, we will surely add it in our project so that code in any language can be tested. •
- **Dashboard:** Dashboard will be added in future where the developer can see their file saved in workspace and can edit their information details.

Theoretical Background

1. **MERN Stack**

   MERN stands for MongoDB, Express.js, React.js, and Node.js, used for making full- stack web applications. We used node.js and express.js for making routes, handling the back-end, and API calls and database connections to the database relying on MongoDB. Frontend is created by React.js and its libraries like GSAP, Framer-motion.

2. **Socket.IO**

   Socket is basically the communication endpoint that enable the devices to establish the connection between the clients over the network. Socket.IO is the JavaScript library that our team used to implement sockets using the socket Id. Real time event-based bidirectional communication between developers on Code doodle to change the code in real time so that updation reflect rapidly to every developer in the room.

3. **WebRTC**

   WebRTC is the open-sourced technology that supports two-way communication with each other in real time using audio, videos. Our team used it in conducting face-to-face meetings between developers to make the environment professional.

4. **Gemini API**

   API is an application Programming Interface that can be used to get a response using the HTTPS header method. The response usually comes in JSON format. We used Gemini API for code-reviewer purpose so that a developer can debug his code and optimize it.

5. **JWT Authentication**

   JWT stands for Json Web Token, which securely transfers information between client and server. We used JWT in token-based authentication to handle validation and verification.

Literature Review

Although tools and platforms exist that support features like coding collaboration, Codeshare, or face-to-face meeting capability, Google meet, no one had provided the solutions in a single platform whatever the developer faces, and switching to these websites separately for different purposes reduces the focus to be applied on the actual code.

Following are the tools that provide their respective solutions:

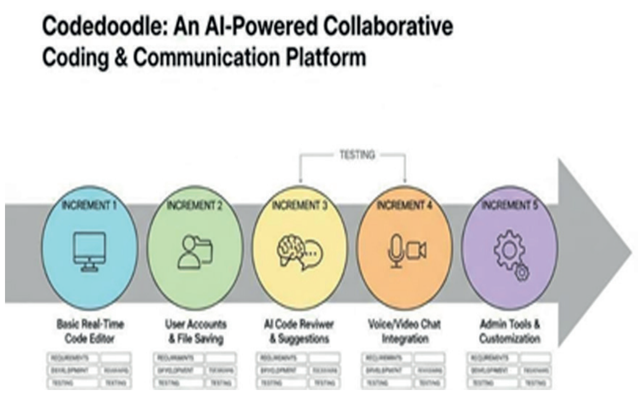
1. **Collaborative Editors:** There are platforms, such as Codeshare, that only provide the code sharing panel to change the code in real time and nothing extra.
2. **Online IDEs:** Replit, Code Pen etc. allow coding and limited collaboration, but they also do not integrate other tools that we tried to include in our project, like video chat and AI- powered code review tools.
3. **Video Conference tools:** Tools like Zoom, Google Meet support communication but lack in providing the code editor to provide the Collab panel.
4. **AI Coding Assistant:** AI-powered tools, like Gemini, GitHub Copilot, and ChatGPT can all generate and analyze code, though not offer collaborative editing and built-in communication.

Literature Gap:

"Research and solution find a gap that there is no single platform which combines Real- time coding, AI review, Video communication, and chat in one web application. This gap is filled by Code doodle so that a developer only focuses on writing code rather than switching between the web applications for a specific need."

Research Methodology

- **Methodology used:** We implemented an incremental model for preparing the code doodle. Where out the team analyzes bugs and other features that are forgot to be added by considering the feedback from people, users, friends, and asking through survey check the drawback if any then try to add the new feature in the new cycle of incremental model to make the website more supportive enough to solve all the problems in a single platform.

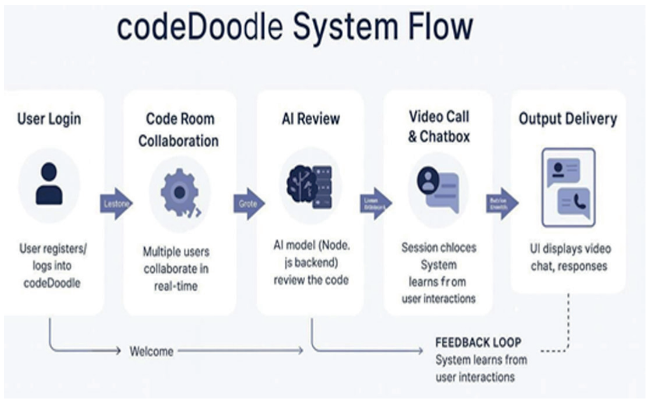

Codedoodle: An AI-Powered Collaborative Coding & Communication Platform
TESTING
INCREMENT 1
INCREMENT 2
INCREMENT 3
INCREMENT 4
INCREMENT 5
Basic Real-Time Code Editor
User Accounts & File Saving
AI Code Reviwer & Suggestions
Voice/Video Chat Integration
Admin Tools & Customization

- **Data collection method:** Generally, we have been posting on the LinkedIn platform for motivating them to use our web application and ask for feedback, and accordingly take the next action to take rightly. Hence, both functional and non-functional requirements were gathered through:
  - i. Social Media Feedback
  - ii. Literature review of existing similar application and identifying the drawback of those and implementing the same in our website.
- **Technology Stack Selection:** MERN stack is chosen because React provides Component architecture, Node.js creates a scalable backend server, and MongoDB stores user details in flexible JSON-like documents.
- **Database Design :** The database that Code doodle supported is achieved using collection, documents and flexible schema. MongoDB is used for performing CRUD operation following is the schema demo that our website is using.
- **Deployment :** The web app was deployed using Vercel and render. Render is designed to make building and running applications and servers easy. It automated our build process from our code repository. Vercel is used - A cloud platform used to deploy the Client folder that contains the Frontend frameworks.

Our backend and frontend deployed link is given below.

Frontend: https://code-doodle-editor.vercel.app/ Backend : https://code-doodle-editor-6.onrender.com/

System Flow Chart


codeDoodle System Flow
User Login
User registers/ logs into codeDoodle
Lestone
Code Room Collaboration
Multiple users collaborate in real-time
Grote
AI Review
AI model (Node. js backend) review the code
Video Call & Chatbox
Session chloces System learns from user interactions
Output Delivery
UI displays video chat, responses
Welcome
FEEDBACK LOOP
System learns from user interactions

- **User Login**

  The user signs up or logs in on the code doodle platform. Authentication is handled here (email/password, etc.)
- **Code Room collaboration**

  The users then join a collaboration room after logging in. Multiple users can write code together, edit code in real time, instantly see each other's changes.
- **AI Review**

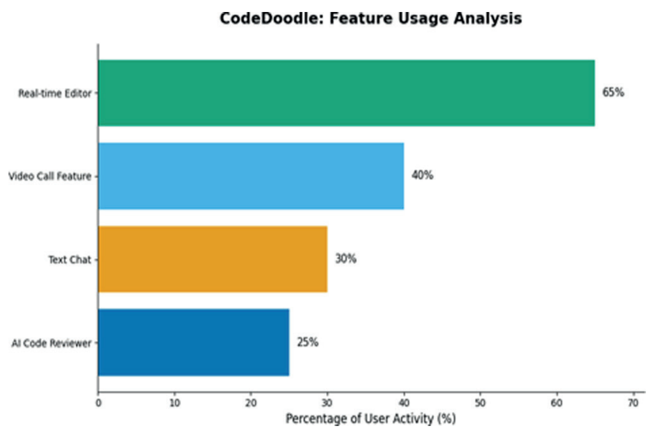
  Code is written and sent to the AI model via the Node.js backend and the AI engine. AI performs code review, error detection, suggestions for improvement.
- **Video Call & Chatbot**

  Users can join a video call inside the platform to discuss code, get feedback, hold group sessions.
- The chatbot is available for quick conversation.
- **Delivery of Output**

  Final output-review of results and responses, video chat interface-is shown. Users can view code review responses, see video sessions, read chat messages.
- **Feedback Loop**

  It acquires information from users' choices and interactions.

Usage analysis of Project



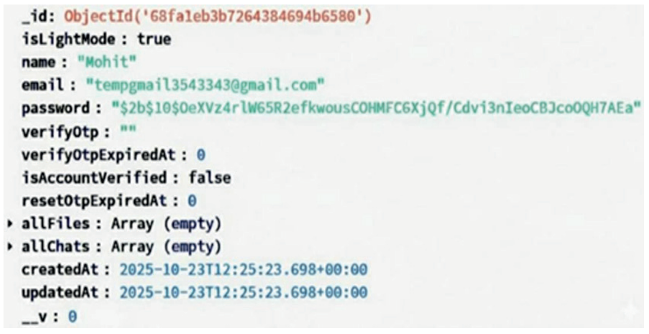
We shared the project with many of our friends and fellow developers. They all said that having every tool in one place was a huge benefit, as nothing like it really existed for them.

From their feedback and use, we learned what features they found most useful. The graph below represents the weightage they gave to each of them:

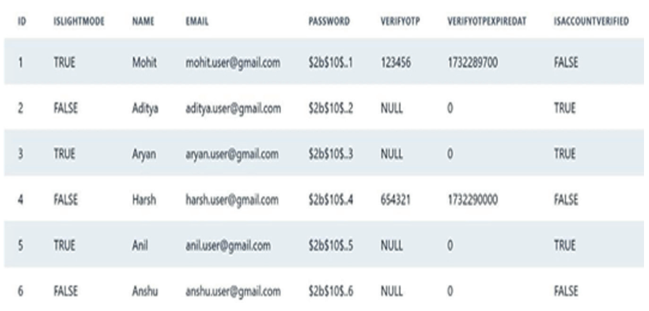
- The Real-time Editor is clearly the most popular, at 65%. This shows that our main goal—fixing "merge hell" and isolated coding—is what users value most.
- Video Call Feature: The second most used feature is responsible for 40 percent, showing how much developers rely on face-to-face communication.
- Other very useful tools that help collaborative work of teams and improvement of their code are the Text Chat (30%) and AI Code Reviewer (25%).

Result and Testing

MongoDB result and Collection Structure after Registering



```
  _id: ObjectId('68fa1eb3b7264384694b6580')
  isLightMode : true
  name : "Mohit"
  email : "tempgmail3543343@gmail.com"
  password : "$2b$10$OeXVz4rlW65R2efkwousCOHMFC6XjQf/Cdvi3nIeoCBJcoOQH7AEa"
  verifyOtp : ""
  verifyOtpExpiredAt : 0
  isAccountVerified : false
  resetOtpExpiredAt : 0
▸ allFiles : Array (empty)
▸ allChats : Array (empty)
  createdAt : 2025-10-23T12:25:23.698+00:00
  updatedAt : 2025-10-23T12:25:23.698+00:00
  __v : 0
```



ID	ISLIGHTMODE	NAME	EMAIL	PASSWORD	VERIFYOTP	VERIFYOTPEXPIREDAT	ISACCOUNTVERIFIED
1	TRUE	Mohit	mohit.user@gmail.com	$2b$10$..1	123456	1732289700	FALSE
2	FALSE	Aditya	aditya.user@gmail.com	$2b$10$..2	NULL	0	TRUE
3	TRUE	Aryan	aryan.user@gmail.com	$2b$10$..3	NULL	0	TRUE
4	FALSE	Harsh	harsh.user@gmail.com	$2b$10$..4	654321	1732290000	FALSE
5	TRUE	Anil	anil.user@gmail.com	$2b$10$..5	NULL	0	TRUE
6	FALSE	Anshu	anshu.user@gmail.com	$2b$10$..6	NULL	0	FALSE

- **ID :-** Unique ID for each Record act as a Primary Key to identiy the record uniquely.
- **IsLightMode:-** Tells where it is LightMode or DarkMode(False) that the user is currently using.
- **Name :-** User registered to Code doodle.
- **Email :-** Email that is used to register the User account.
- **Password :-** Encrypted Password for that Registered Email.
- **VerifyOTP :-** OTP used to verify the Email to use the Tools of Code doodle.
- **VerifyOTPExpiredAt :-** Tells the time duration within which the OTP is valid for verification of user account.
- **IsAccountVerified :-** Check whether the Account is verified or not.

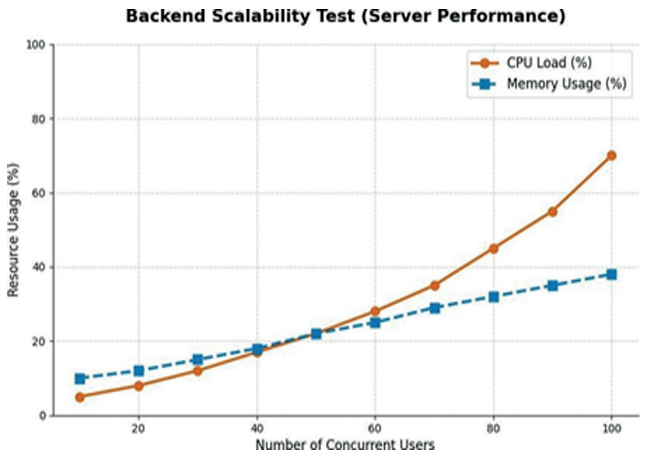
After building the system, many tests were conducted to verify its correctness, performance, and reliability. The objective of the testing phase was to ensure that all functional requirements were met and that the system performed good under different conditions.

Functional Testing

- Functional testing ensures that every feature of code doodle works as expected.
- This includes testing user login, code room collaboration, AI review, and video/chat features.
- This means assuring that the system performs its intended work and behaves as expected.

Scalability Testing

- Performance testing verifies how fast and stable code doodle works under a variety of loads.
- Multiple users were simulated in a shared code room to observe system response time.



Unit Testing

- Unit testing tests each separate, individual component of code doodle in isolation.
- Login validation, logic of code execution, AI review functionality, and rendering of UI were tested separately in separate modules.
- Expected and unexpected inputs were used to test each module.
- This helped identify bugs early before combining modules together.

Integration Testing

- Integration testing checks the interaction of various modules among themselves in code doodle.
- It checks the flow of data between login → collaboration rooms → AI review → video/chat → output.
- APIs and backend functions were tested for smooth communication.

Socket Testing

- Socket testing was performed because code doodle supports real-time collaboration and live updates.
- It ensures socket creation, maintenance, and closure are performed correctly.
- This testing helps to ensure that changes in code reflect back to every developer in the room smoothly.

API Testing

- API testing ensures that all backend calls in code doodle return correct and consistent responses.
- The APIs tested included Login API, room creation API, code send/receive API, and AI review API.
- This testing is done by Postman Application. It is an application where we can manage, test and handle the API response effectively.

Resources and advantages and disadvantages

Hardware Resources

- Cloud servers for hosting backend services, and real-time collaboration.
- Storage for user data, code snippets, and session logs.
- Developer machines for coding, testing, and deployment.

Software Resources

- Frontend: React/Next.js, HTML, CSS, JavaScript.
- Backend: Node.js, Express.js, WebSocket server.
- Database: MongoDB for storing users, rooms, chats.
- Video/Chat Tools: WebRTC for real-time communication.
- Version Control: GitHub.

Human Resources

- Full-stack developer for frontend + backend.
- UI/UX designer for clean interface.

Network & Connectivity Resources

- Stable internet for face-to-face communication and live-time syncing of code To ensure that code remain consistent.
- Load balancer for scalability.

Advantages

- **One Place Everything**

  There is no single platform which combines Real- time coding, AI review, Video communication, and chat in one web application but the gap is filled by Code doodle.
- **Smooth User Workflow**

  Login → Code Room → AI Review → Video/Chat → Output Delivery is seamless and user-friendly.
- **Cloud-based Platform**

  No installations required; users can access the system from anywhere with just a browser

Disadvantages

Complexity in Implementation

Managing sockets, video calling, and AI integration together can be technically challenging.

High Server Cost

Real-time collaboration, AI processing, and WebRTC require strong stun servers, which is usually high cost.

Internet Dependency

System performance decreases with slow internet, affecting video and real-time collaboration quality and speed as well.

AI Limitations

AI code review may not always be 100% accurate or context-aware, requiring manual verification.

Only Support Peer-to-Peer connection

WebRTC support only peer-to-peer connection and our team used the same in Code Doodle.

Conclusion

Code doodle effectively integrates chat, video communication, AI-powered code review, and real-time collaborative coding into one single platform. It creates an environment in which communication and coding can flow without necessarily having to operate a number of apps by bringing these features together. This makes Code doodle a unique platform to all types of people for professionals, students, and developers who work from home or collaborate remotely.

Consequently, one of the biggest advantages of Code doodle is that users will not have to navigate to different websites for chatting, video calling, coding, reviewing, and sharing the work. Instead of using one website to collaborate on code, another for video meetings, and another tool altogether to get AI feedback, everything is put into one hub. This eliminates unnecessary navigation, reduces the amount of time wasted by changing app for different purposed.

Thus, by acting as an all-in-one platform, Code doodle considerably enhances productivity and efficiency. It also enhances learning for students and simplifies the workflow of remote developers.

On the whole, Code doodle is a very convenient collaboration tool. It not only connects communication and coding but also strengthens them through real-time updates and AI- supported review feedback. Capable of replacing several platforms with one feature-rich environment, which increased productivity to modern digital teamwork.

References

1. Banks, A., & Porcello, E. (2017). Learning React: Functional Web Development with React and Redux. O'Reilly Media.
2. Cannelton, M., Harter, M., Toloache, T., & Rajish, N. (2014). Node.js in Action. Manning Publications.

3. Williams, J. D., & Mahmood, Z. (2018). Cloud Computing: Methods and Practical Approaches. Springer.
4. Loreto, S., & Romano, S. P. (2014). Real-Time Communications in Web Browsers with WebRTC. IEEE Internet Computing, 18(5), 66–73.
5. Kleppmann, M., & Beresford, A. R. (2017). A Conflict-Free Replicated JSON Datatype. IEEE Transactions on Parallel and Distributed Systems, 28(10), 2733– 2746. Used in real-time editors such as Google Docs.
6. Nguyen, T., & Nguyen, H. (2020). An Evaluation of Real-Time Collaboration Tools for Software Development. International Journal of Computer Applications, 176(33), 1–7.
7. Melo, F., & Pereira, A. (2021). Analysis of AI-assisted Programming Tools in Software Engineering Education. ACM SIGCSE, 150–158.
8. React.js Official Documentation. (n.d.). Retrieved from https://react.dev
9. Node.js Official Documentation. (n.d.). Retrieved from https://nodejs.org
10. MongoDB Atlas Documentation. (n.d.). Retrieved from https://www.mongodb.com/docs/
11. Socket.IO Documentation. (n.d.). Retrieved from https://socket.io/docs
12. WebRTC Official Guide. (n.d.). Retrieved from https://webrtc.org/getting-started
13. Google Gemini API Documentation, n.d. Available at: https://ai.google.dev
14. Tailwind CSS Documentation. (n.d.). Retrieved from https://tailwindcss.com/docs
15. GSAP (Green Sock Animation Platform). (n.d.). Retrieved from https://gsap.com/docs
16. Online Articles (Credible Technical Sources)
17. Richardson, L. (2021). A Beginner's Guide to JWT Authentication. Auth0 Engineering Blog. https://auth0.com/blog
18. Singh, A. (2023). Implementing Real-Time Applications Using Socket.IO.
19. Digital Ocean Community. https://www.digitalocean.com/community
20. Brown, C. (2022). Building Video Chat Apps Using WebRTC and Node.js.
21. Log Rocket Blog. https://blog.logrocket.com
22. Impact of COVID-19 on Software Development https://ieeexplore.ieee.org/document/9453143 Additional Credible Technical Sources
23. Mozilla Developer Network [MDN]. (n.d.). WebRTC API. https://developer.mozilla.org/en-US/docs/Web/API/WebRTC_API
24. Google Open Source. (2024). WebRTC Tracing and Architecture Overview. https://opensource.google