Assignment: CodeMate – Real-Time Collaborative Coding Platform with Voice Support

Problem Statement

In most online learning environments, students struggle to clearly explain their coding issues over chat. This leads to delays, back-and-forth messages, requests for screen shares or GitHub links, and overall inefficiency in doubt resolution.

CodeMate aims to solve this problem by providing a real-time collaborative coding environment where students and teaching assistants (TAs) can:

- Code together live
- Talk via voice chat
- Share errors and outputs instantly
- Store and revisit previous sessions

Your task is to design and develop **CodeMate**, a platform that enhances the mentoring experience by eliminating communication friction and making it easier to debug and learn code collaboratively.

Key Features to Implement

1. **Authentication & User Roles**
    - Implement **JWT-based authentication** (Login/Register)
    - Roles: `Student` and `Teaching Assistant (TA)`
    - TAs have access to more features (e.g., dashboard, join any session)
2. **Session Rooms**
    - Users can **create or join a coding session** using a unique Room ID or shareable link

- Sessions should persist even after users leave, so they can revisit the session later

3. **Real-Time Collaborative Editor**
   - Use **CodeMirror** or **Monaco Editor** (like VS Code)
   - Sync code between users in real-time using **WebSockets** (Socket.IO or tRPC)
   - Support **syntax highlighting** and **multiple programming languages**: Python, C++, Java

4. **Live Code Execution**
   - Use **Judge0 API** or a **Docker-based sandbox** to run code
   - Display the **output or errors** in real-time to all users in the session

5. **Voice Calling**
   - Integrate **1:1 voice support** using **WebRTC** (via PeerJS, Agora, or Daily.co)
   - This allows students and TAs to speak and resolve doubts instantly

6. **Persistent Session Storage**
   - Save the **code history**, timestamps, and participants to a **MongoDB** or **Firebase** backend
   - Users should be able to revisit previous sessions like a versioned document

7. **Raise Hand & TA Dashboard**
   - Students can click “**Raise Hand**” to notify a TA for help
   - A special **dashboard for TAs** shows all active sessions with student names and raised hands

8. **Additional Features (Optional / Bonus)**
   - Inline code comments or text chat inside the editor
   - Session history viewer (similar to Google Docs versioning)

    - Theme toggle (light/dark)
    - Mobile responsiveness

Recommended Tech Stack

Frontend

- **React.js**
- **CodeMirror** or **Monaco Editor**
- **WebRTC** integration (PeerJS, Agora, or Daily.co)
- WebSocket support (Socket.IO or tRPC client)

Backend

- **Node.js with Express.js**
- **Socket.IO** or **tRPC** for real-time sync
- **WebRTC signaling server** if using PeerJS
- **Code execution**: Judge0 API or custom Docker-based execution engine

Database

- **MongoDB** (with Mongoose) or **Postgres (Prisma)**
- Store session metadata, code history, user info, and raised hand status