Video Calling Web Application

Objective

Design and implement a video calling web application that allows users to create and join video calls, chat, and manage audio/video settings. The application should use **WebRTC** for real-time communication and should provide a responsive, user-friendly interface.

Features to Implement

Core Features

1. **Create Video Call Meeting**
    - A user should be able to create a new video call meeting.
    - A unique meeting ID or link should be generated for each meeting.

2. **Join Video Call**
    - Users should be able to join a video call using the meeting link or ID.
    - Multiple users should be able to join the same meeting (supporting more than 2 participants).

3. **Chat Functionality**
    - Users in a video call should be able to send text messages to each other in real-time.

4. **Audio/Video Toggle**
    - Users should have options to mute/unmute their microphone.
    - Users should be able to toggle their camera on/off.

5. **Screen Sharing**
    - Users should be able to share their screen with other participants in the call.

Optional Features (Advanced)

1. **Admin Controls**
    - Admin (meeting creator) can:
        - Grant permission to users to unmute or toggle video.
        - Allow or deny users from entering the meeting.

        - Grant screen share permissions to participants.

2. **Schedule Future Meetings**
    - Users should be able to schedule a meeting at a specific date and time.
    - The scheduled meeting details (title, date, time, meeting link) should be saved in the database.
    - Provide a "My Meetings" section where users can view upcoming scheduled meetings.
    - (Bonus) Send an email/notification reminder before the meeting starts.

UI/UX Requirements

- The app should have a **clean and responsive UI**.
- Use modern design principles, clear buttons for audio/video, chat, and screen share.
- Responsive design for desktop and mobile screens.
- Optional: Dark/light mode toggle for accessibility.

Technical Requirements

- **WebRTC must be used** for video/audio communication.
    - ❌ Use of third-party communication services such as *100ms, Agora, Twilio, Jitsi, Vonage, etc.* will **not be considered**.
- **Tech Stack Recommendation:** MERN (MongoDB, Express, React, Node.js).
    - ✅ Recommended, but **not mandatory**. You can use other stacks if preferred.
- **Signaling and Chat:** Socket.IO (or equivalent real-time library).
- **Database:** Any database (MongoDB recommended).
- **Deployment:** The app must be deployed and accessible online (e.g., Vercel, Netlify + Render, Heroku, AWS).

Bonus Points

- Implement **meeting recording** feature.
- Add **notification sounds** for user join/leave or messages.
- Implement **password-protected meetings** for security.
- Implement **React context or Redux** (if using React) to manage global state effectively.

Deliverables

1. Source code in a Git repository.
2. **Deployed link of the application (mandatory).**
3. Instructions to run the project locally.
4. Screenshots or a demo video showing:
    - Meeting creation
    - Joining a call
    - Chatting
    - Audio/video toggle
    - Screen sharing
    - (Optional) Admin controls + scheduled meetings

Evaluation Criteria

- Correct implementation of video/audio calls using **WebRTC**.
- Real-time chat functionality.
- UI/UX quality.
- Proper code structure and documentation.
- Deployment accessibility (link must work).
- Optional features (like admin controls, scheduled meetings) will be considered for bonus marks.