Technology Stack – *Video Conferencing Web App*

🔷 1. Architectural Overview

This application follows a **scalable 3-tier architecture**:

Presentation Layer → Application Logic Layer → Data & Media Layer

(React.js Frontend) (Node.js, Socket.io) (MongoDB, Agora SDK)

- **Frontend (Client)**: UI/UX interface for video, chat, and controls
- **Backend (Server)**: Authentication, room management, real-time messaging
- **Media Server / SDK**: Real-time video/audio stream via Agora

🔷 2. Component-Wise Technology Breakdown

Component	Description	Technology Used
🌐 **Frontend (UI)**	User-facing interface	React.js, HTML/CSS
⚙️ **Backend**	Business logic, API routes, authentication	Node.js, Express.js
🔁 **Real-Time Engine**	Real-time signaling and chat	Socket.io
🎥 **Media SDK**	Video & audio communication	Agora Web SDK
🧩 **Authentication**	User sign-up, login, token verification	JWT / Firebase Auth
🗄️ **Database**	Storage of user data, room sessions, chat history	MongoDB (MongoDB Atlas for cloud hosting)
🧪 **Testing**	Component and API testing	Jest, Postman, React Testing Library
📡 **APIs / External**	SDK integrations and analytics	Agora APIs

🔷 3. Development Tools

Tool	Purpose
Visual Studio Code	Code editor
Postman	API testing
Git & GitHub	Version control & collaboration
Figma / Canva	UI mockups (optional)

🔷 4. Application Characteristics

Characteristic	Description
Open Source	Uses open frameworks like React, Node.js, and Socket.io
Modular Design	React components, Express routes, and service-based logic