Solution Requirements – *Video Conferencing Web App*

🔹 1. Functional Requirements

These define what the system should do — the **core features** offered to users.

FR No.	**Epic / Feature**	**User Stories / Sub-Tasks**
FR-1	**User Authentication**	- Sign up, Login, Logout, Password Reset - OAuth login (Google / GitHub)
FR-2	**Meeting Creation & Joining**	- Host creates a new meeting room - Generates a unique room code/link
FR-3	**Video & Audio Communication**	- Real-time media streaming via Agora - Mute/unmute audio - Enable/disable video
FR-4	**Real-Time Chat**	- Text-based messaging during the call - Socket.io-based event handling
FR-5	**Participant Management**	- View participant list - Host can remove or mute participants
FR-6	**Screen Sharing**	- Any user can share their screen (browser screen sharing API + Agora integration)
FR-7	**Meeting Exit & End**	- Users can leave anytime - Host can end the session for all participants

🔷 2. Non-Functional Requirements

These define the **quality attributes** of the system — how it should perform.

NFR No.	Requirement	Description
NFR-1	**Usability**	UI should be intuitive, minimalistic, responsive across devices
NFR-2	**Security**	- Use HTTPS - Token-based authentication - Role-based access control
NFR-3	**Performance**	- Page load < 2 seconds - Video latency < 300ms - Instant chat delivery
NFR-4	**Reliability**	System should function under different network conditions with graceful fallback
NFR-5	**Availability**	Maintain 99.9% uptime through cloud deployment
NFR-6	**Scalability**	Able to handle multiple simultaneous meetings and users via Socket.io rooms & Agora channels

✅ Summary

This solution is designed to be **modular, real-time, and secure**, ensuring a smooth user experience whether you are hosting or joining a meeting. Core technologies like **React**, **Node.js**, **Socket.io**, and **Agora** are used to achieve real-time communication, supported by **MongoDB** and **JWT-based authentication**.