Project Design Phase

# Proposed Solution Template

Date: 28-05-2025

Project Name: Video Conferencing App

Maximum Marks: 2 Marks

|  |  |
| --- | --- |
| Parameter | Description |
| Problem Statement (Problem to be solved) | Many users lack access to simple, secure, and fast video conferencing tools that do not require third-party platforms or installations. Existing solutions are either expensive, require sign-ups, or are bloated with non-essential features. There is a growing need for a lightweight, privacy-focused alternative. |
| Idea / Solution Description | This project delivers a full-stack, browser-based video conferencing application using WebRTC and PeerJS for direct peer-to-peer audio/video communication. The system allows users to create/join rooms, share screens, and exchange chat messages with no download or account creation required. The signaling server is built with Node.js and Socket.IO for lightweight coordination. |
| Novelty / Uniqueness | - 100% browser-native with no installation required - No user registration/login for quick access - Peer-to-peer encrypted media with no server relay - Chat integrated with live sessions - One-click screen sharing - Recording functionality (downloadable session) - Clean, minimal UI with custom room IDs and dynamic links - No third-party dependencies for media management - Easily embeddable into other apps or platforms |
| Social Impact / Customer Satisfaction | - Free and open alternative to Zoom/Meet for education and community use - Enhances accessibility in low-resource environments - Provides private, account-less communication - Useful for NGOs, student groups, quick team calls, and telehealth - Reduces digital divide by eliminating sign-up friction - Empowers developers to build secure video platforms with ease |
| Business Model (Revenue Model) | - Open-source community version with premium hosted SaaS option - Paid tier with cloud-based call recording, analytics dashboard - White-label offering for schools, startups, and enterprises - Partner integrations (e.g., calendar/scheduler API access) - Developer support plans and hosted APIs for quick adoption |
| Scalability of the Solution | - Uses lightweight signaling server: minimal backend load - Peer-to-peer model scales naturally with more users - TURN/STUN services can be added for NAT traversal - Modular code structure for future feature expansion (e.g., breakout rooms, live polling) - Easily deployable to cloud/VPS using Docker or PM2 - Can integrate WebSockets, Redis, or Kafka for broader feature support |