

Phase-wise Documentation: Video Conferencing App

Phase 1: Requirement Gathering

- User Registration & Login (Authentication)
- Meeting Room Creation
- Unique Meeting Room Links
- Video & Audio Communication (using Agora/WebRTC)
- Real-time Chat (Socket.io)
- Participant Management
- Responsive Frontend (React)
- Backend API (Node.js + Express)
- Database (MongoDB)
- Room Management with Context API

Phase 2: Project Setup

Frontend:

- React project setup with create-react-app
- Install dependencies: react-router-dom, axios, socket.io-client, agora-rtc-react

Backend:

- Node.js + Express project setup
- Install dependencies: express, mongoose, cors, socket.io, bcrypt, jsonwebtoken, dotenv

Version Control:

- Git repository initialized

- .gitignore configured

Phase 3: Authentication Module

Backend:

- User Model (MongoDB)
- Auth routes for Register & Login
- Password Hashing with bcrypt
- JWT Token Generation

Frontend:

- AuthContext using React Context API
- Login & Register Pages
- Token-based Login Flow
- Protected Routes for Meeting Pages

Phase 4: Meeting Room Creation

Backend:

- API to create & fetch meeting rooms
- Store meeting details in MongoDB

Frontend:

- Room Creation Page
- Display Unique Room Links
- Navigate to Room using ID
- Room Join Validation

Phase 5: Real-Time Communication

Socket.io Integration:

- Establish Socket connection on joining room
- Manage room participants
- Real-time chat within room

Agora/WebRTC Integration:

- Video & Audio streaming setup
- Display participants' video
- Handle join/leave events

Phase 6: UI/UX Development

- Clean, responsive design using CSS/React
- Meeting Room Layout:
 - Video grid
 - Chat section
 - Participants list
 - Leave button
- Navbar with navigation

Phase 7: Error Handling & Validation

- Input Validation (Register/Login forms)
- Room Join validation
- Connection error handling
- Loading states for API calls

Phase 8: Deployment

Backend:

- Deploy to Render/Heroku/Vercel
- Environment variables configured

Frontend:

- Build React app
- Deploy to Netlify/Vercel

Phase 9: Testing

- Functional Testing:
 - Login, Register
 - Room Join/Leave
 - Video/Audio working
 - Chat functionality
- Basic security checks:
 - Token validation
 - Unauthorized access prevention

Phase 10: Documentation & Final Touches

- README with project overview, setup steps
- API Documentation (brief)
- Folder Structure Explanation
- .env Example file
- Future Improvements listed

Optional Future Phases:

- Screen Sharing
- Recording Feature
- Raise Hand / Mute All
- Admin Controls for rooms
- Dockerize the project