## **📅 2-Week Daily Task Plan**

⏱️ Assume ~6–7 hrs/day. Each day builds on the previous.

### **Week 1 — Core Backend + Auth + Realtime Basics**

**Day 1: Planning & Setup**

* Project structure (frontend + backend folders)
* Initialize Git repo
* Install dependencies (Node/Express, MongoDB, React, Tailwind, etc.)
* Design database schema (Users, Workspaces, Documents, Tasks, Messages)

**Day 2: User Auth System**

* Create signup/login APIs (JWT + bcrypt)
* Connect MongoDB Atlas
* Create basic user model
* Build frontend auth pages (login/signup)

**Day 3: Workspaces & User Management**

* Workspace model (name, members)
* APIs for create/join workspace
* Workspace dashboard UI (after login)

**Day 4: Real-time Chat Setup**

* Add Socket.io server
* Create chat model
* Build workspace-level chat rooms
* Realtime message updates

**Day 5: Task Board System**

* Create task board model (lists, tasks)
* APIs for CRUD tasks/lists
* UI with react-beautiful-dnd
* Sync tasks in real-time with Socket.io

**Day 6: Document Collaboration Setup**

* Integrate Yjs with TipTap/Quill editor
* Set up Y-WebSocket server
* Store document snapshots in MongoDB

**Day 7: Document Editor Frontend**

* Create document editor page
* Implement multi-user real-time editing
* Test syncing between multiple tabs/users

### 

### 

### **Week 2 — Video, Whiteboard, Polishing & Deployment**

**Day 8: WebRTC Video Conferencing**

* Setup PeerJS or simple-peer signaling over Socket.io
* Implement 1:1 and group video calls
* UI for call popup inside workspace

**Day 9: Collaborative Whiteboard**

* Build whiteboard with Konva.js
* Sync drawing actions via Socket.io
* Add clear, undo, export image

**Day 10: Workspace UI Integration**

* Combine chat, tasks, docs, whiteboard, video into one workspace UI
* Tabbed or multi-panel interface

**Day 11: File Uploads + User Profiles**

* Cloudinary integration
* Profile pictures
* Workspace file sharing

**Day 12: Notifications & Presence**

* Show online/offline users
* New message/task/doc change notifications
* Email invites to join workspace

**Day 13: Testing & Bug Fixing**

* Write API tests (Jest/Postman)
* Resolve UI and performance issues
* Polish UI styling with Tailwind

**Day 14: Deployment & README**

* Deploy frontend (Vercel) and backend (Render/Railway)
* Seed demo data
* Write full README (setup + features + screenshots + demo link)