**YouTube**, Reddit**, Netflix, WhatsApp, Flipkart, Discord, and Zoom**—basically an **all-in-one super app**. 🔥

**📅 Timeline (6 hours/day)**

**Total Time:** **~8-10 weeks** (~2-2.5 months)

**🚀 Phase 1: Setup & Core Features (Week 1-2)**

🔹 **Day 1-2** → Project setup (**Flutter, Firebase, Node.js, MongoDB**)  
🔹 **Day 3-5** → User authentication (**Sign-up/Login, Firebase/Auth0**)  
🔹 **Day 6-8** → Post System (**Reddit-style posts, images, text, upvotes**)  
🔹 **Day 9-12** → Backend API for posts (**MongoDB/PostgreSQL + Express.js**)  
🔹 **Day 13-14** → Comments, upvotes, notifications

✅ **Goal:** Users can register, log in, create posts, comment, and upvote.

**📨 Phase 2: Messaging & Real-Time Chat (Week 3)**

🔹 **Day 15-16** → Chat UI (**like WhatsApp/Discord**)  
🔹 **Day 17-18** → Real-time messaging (**WebSockets/Firebase Firestore**)  
🔹 **Day 19-20** → Group chat, seen/unseen messages

✅ **Goal:** Real-time messaging system is working.

**📹 Phase 3: Video Upload & Streaming (Week 4-5)**

🔹 **Day 21-22** → UI for uploading videos  
🔹 **Day 23-24** → Backend for video storage (**Firebase Storage / AWS S3**)  
🔹 **Day 25-26** → Convert videos for streaming (**FFmpeg, HLS**)  
🔹 **Day 27-28** → Implement a video player (**like YouTube**)  
🔹 **Day 29-30** → Likes, comments, subscriptions (like YouTube)  
🔹 **Day 31-32** → Improve video recommendations

✅ **Goal:** Users can **upload, watch, and interact with videos**.

**📹 Phase 4: Live Streaming (Optional) (Week 6)**

🔹 **Day 33-34** → Setup WebRTC/RTMP for live streaming  
🔹 **Day 35-36** → Add real-time chat for live videos

✅ **Goal:** Users can **live stream with a chat feature**.

**🎥 Phase 5: Video Calls (Like Zoom) (Week 7)**

🔹 **Day 37-38** → UI for video calling  
🔹 **Day 39-40** → Implement WebRTC or Agora SDK  
🔹 **Day 41-42** → Group calls, screen sharing (optional)

✅ **Goal:** Users can **make video calls**.

**🛒 Phase 6: E-commerce (Week 8)**

🔹 **Day 43-44** → Product listing UI (**like Flipkart**)  
🔹 **Day 45-46** → Cart system (local storage/backend)  
🔹 **Day 47-48** → Payments (**Stripe, Razorpay**)  
🔹 **Day 49-50** → Order management

✅ **Goal:** Users can **buy & sell products**.

**🎨 Phase 7: UI/UX Polish & Deployment (Week 9-10)**

🔹 **Day 51-53** → Dark mode, animations, UI fixes  
🔹 **Day 54-56** → Backend deployment (**Render, Vercel, AWS**)  
🔹 **Day 57-58** → Build & test Android/iOS/Desktop apps

✅ **Goal:** **Fully deployed** multi-platform app.

**⏳ Final Breakdown**

| **Phase** | **Time Estimate** |
| --- | --- |
| Core Features (Auth, Posts) | **2 weeks** |
| Messaging (Chat System) | **1 week** |
| Video Upload & Streaming | **2 weeks** |
| Live Streaming (Optional) | **1 week** |
| Video Calls (Zoom-like) | **1 week** |
| E-commerce & Payments | **1 week** |
| UI Polish & Deployment | **2 weeks** |
| **Total Time** | **8-10 weeks (~2-2.5 months)** |

**🔥 Tech Stack for This Super App**

| **Component** | **Tech** |
| --- | --- |
| **Frontend (Cross-Platform)** | Flutter (Dart) |
| **Backend (Server)** | Node.js (Express.js, Nest.js) |
| **Database** | MongoDB (NoSQL) or PostgreSQL (SQL) |
| **Real-Time Chat** | WebSockets (socket.io) / Firebase Firestore |
| **Video Uploads** | Firebase Storage / AWS S3 / Cloudflare R2 |
| **Video Streaming** | FFmpeg + HLS (like YouTube/Netflix) |
| **Live Streaming** | WebRTC (for low-latency streams) |
| **Video Calls** | WebRTC / Jitsi / Agora SDK |
| **E-commerce** | Stripe / Razorpay |

**Final Thoughts**

✅ **Focus on an MVP** first, then expand.