**WhatsApp Clone - Full Module Plan**

**📅 Overview:**

Build a WhatsApp Clone as a full-stack web application using:

* **Frontend**: React.js
* **Backend**: Node.js + Express.js
* **Database**: MongoDB + Mongoose
* **Realtime Messaging**: Socket.IO
* **Calling**: WebRTC

**✅ Full List of Modules (Each in Detail)**

**🔀 Module 1: Project Initialization & Setup**

**Goal:** Prepare the base project structure and required tools

* Setup folder structure: client/ for React, server/ for Node
* Install and configure:
  + Node.js + Express
  + MongoDB + Mongoose
  + Socket.IO server
  + React with Router, Axios, socket.io-client
* Add environment variables and testing endpoints

**👤 Module 2: User Authentication & Authorization**

**Goal:** Implement user login and registration system

* **Backend:**
  + User schema with hashed passwords (bcrypt)
  + JWT-based authentication system
  + Routes: /api/register, /api/login, /api/user
* **Frontend:**
  + Register and Login forms
  + Auth context or Redux store
  + Store and validate JWT token

**🛋️ Module 3: Chat System Backend**

**Goal:** Enable storing and managing conversations/messages

* MongoDB Models:
  + User
  + Message
  + Chat (room)
* Routes:
  + Get all chats for a user
  + Send a message to a chat
  + Create/find a private chat between two users

**📅 Module 4: Frontend Chat UI**

**Goal:** Create user-friendly chat interface

* Sidebar with recent chats
* Chat window with messages
* Scrollable and auto-updating message list
* Input box with emoji and send button

**🌐 Module 5: Realtime Messaging with Socket.IO**

**Goal:** Enable live communication between users

* Setup Socket.IO on both frontend and backend
* Connect user to socket on login
* Emit and listen to messages instantly
* Update UI without page refresh
* Typing indicator (optional)

**🚫 Module 6: Private Chats & User List**

**Goal:** Allow chatting only with selected users

* Fetch user list from backend
* Start a private chat by clicking on user
* Prevent duplicate chats with same user
* Highlight active chat

**🧑‍💼 Module 6.5: User Profile Management**

**Goal**: Allow users to view and update their profile info (name, about, profile picture)

**🔧 Backend**

* Extend User schema:

js

CopyEdit

name: String,

phone: String,

about: String,

profilePic: String (image URL),

* Routes:
  + GET /api/user/:id – fetch user profile
  + PUT /api/user/update – update profile info

**🖥️ Frontend**

* Create "Profile" page:
  + Fields: Name, About, Phone (disabled), Profile Picture
  + Upload and preview new profile picture
* Use FormData to send profile pic
* Save pic URL in MongoDB via backend

**📱 Module 6.6: Phone Number Verification**

**Goal**: Register/login users with phone numbers

You can use this instead of email/password or as an additional layer.

**🔧 Backend**

* Add phone number to user schema (unique)
* Route:
  + POST /api/register-phone
  + Integrate with **OTP API** (e.g., Twilio, Firebase)
  + Store verified phone

**🖥️ Frontend**

* Phone login component
  + Input: Phone number
  + Button: “Send OTP”
  + OTP verification screen
* Handle token creation after verification

**🖼️ Module 6.7: Profile Picture Upload**

**Goal**: Let users upload/update their profile picture

**🔧 Backend**

* Use multer to handle image uploads
* Upload to:
  + **Local storage** (for testing)
  + Or **Cloudinary/S3** for production
* Return image URL and save to user model

**🖥️ Frontend**

* Upload section on profile page
* Preview uploaded image
* Show picture in:
  + Sidebar
  + Chat header
  + Status view

**👤 Module 6.8: Rename Chat Contact (Client-side alias)**

**Goal**: Let users rename how they see other users

This is **personal aliasing**, just like WhatsApp lets you name contacts differently.

**🔧 Backend**

* Optional: Create user\_aliases in schema or keep localStorage based

js

CopyEdit

userId: String,

aliases: [{ contactId: String, name: String }]

**🖥️ Frontend**

* Add rename option in chat settings (3-dot menu)
* Save alias locally or persist in DB
* While rendering chat list or messages, show alias name if exists
* **📞 Module 7: Audio Call Feature**

**Goal:** Add one-on-one audio calling functionality using WebRTC

* Setup peer-to-peer audio connection via WebRTC
* Socket.IO signaling for call setup
* Call UI: dial, receive, reject
* Handle microphone access and call end logic

**🎥 Module 8: Video Call Feature**

**Goal:** Implement one-on-one video calling

* WebRTC peer connection with video stream
* Access webcam and microphone
* UI for video grid (caller + receiver)
* Call quality management

**🖼️ Module 9: 24-Hour Status Feature**

**Goal:** Let users post temporary photo/video/text updates (like WhatsApp status)

* Backend:
  + Status schema (userId, media, text, timestamp)
  + Expire status after 24 hours (via TTL index or scheduled job)
* Frontend:
  + Upload status UI
  + View status from other users
  + Carousel-like story viewer

**🎨 Module 10: Media & File Sharing (Optional)**

**Goal:** Enable sending media in messages

* Upload and preview images/videos
* Store media URLs in MongoDB
* Show media in chat bubbles
* Add download/view links for files

**🌍 Module 11: Online Status & Presence**

**Goal:** Show which users are online/offline

* Use Socket.IO to track active users
* Emit user online/offline status
* Show green dot for online users
* Update status in real-time

**📰 Module 12: Notifications (Optional)**

**Goal:** Notify users of new messages

* Message preview in browser tab
* Sound alerts on message receive
* Unread badge count in sidebar
* Desktop notifications API (optional)

**🚀 Module 13: Deployment**

**Goal:** Make your app available online

* Frontend:
  + Deploy to Vercel or Netlify
* Backend:
  + Deploy to Render or Railway
* Database:
  + Host on MongoDB Atlas
* Environment configuration

**🌟 Bonus Ideas (Optional Features)**

* Dark Mode toggle
* Archive chats
* Block/report users
* Delete messages
* Pinned messages
* End-to-end encryption (concept only)

**✅ Final Outcome:**

After completing these modules, you will have a fully functional WhatsApp Clone with:

* Authentication
* Realtime private messaging
* Audio/video calling
* 24-hour status feature
* Media sharing
* Online presence
* Deployment ready for production