Real-Time Chat Application (MERN + WebSockets)

A real-time chat application inspired by WhatsApp / Slack, built using the MERN stack with Socket.IO for real-time communication and MongoDB for message persistence.

Features:
Core Features

- Real-time one-to-one private messaging
- Real-time global & room-based chats
- Persistent chat history using MongoDB
- Image and file sharing
- User authentication (Register & Login)
- Responsive UI (Desktop-first)
- Socket.IO based two-way communication

WhatsApp-like Capabilities

- Left sidebar with chat list
- Right panel with selected chat
- Messages visible only for selected conversation
- Inline image preview (not just download)
- File download support for non-image files

Tech Stack

Frontend

- React.js (Vite)
- Socket.IO Client
- Axios
- HTML, CSS (Flexbox-based layout)

Backend

- Node.js
- Express.js
- Socket.IO
- MongoDB (Mongoose)

Project Structure

```
realtime-chat-app/
│
├── client/                # React frontend
│   ├── src/
│   │   ├── components/
│   │   │   ├── PrivateChatBox.jsx
│   │   │   ├── Sidebar.jsx
│   │   │   └── Message.jsx
│   │   ├── pages/
│   │   │   ├── Login.jsx
│   │   │   ├── Register.jsx
│   │   │   └── Chat.jsx
│   │   ├── services/
│   │   │   ├── api.js
│   │   │   └── socket.js
│   │   └── App.jsx
│   └── package.json
│
├── server/                # Node + Express backend
│   ├── models/
│   │   ├── User.js
│   │   ├── Message.js
```

```
│   │   └── Chat.js
│   ├── routes/
│   │   ├── authRoutes.js
│   │   ├── messageRoutes.js
│   │   ├── chatRoutes.js
│   │   └── uploadRoutes.js
│   ├── socket/
│   │   └── socket.js
│   ├── uploads/            # Uploaded files/images
│   ├── server.js
│   └── package.json
```

Local Setup Instructions

Prerequisites

Make sure the following are installed:

Software	**Version**
Node.js	v18.x or above
npm	v9.x or above
MongoDB	v6.x (local)
Browser	Chrome / Edge

Step-by-Step Guide: From ZIP Download to Local Execution

STEP 1: Download the Project ZIP

1. Download the project ZIP file (shared via email / GitHub / Drive)
2. Save it to your system (e.g., Desktop)

Example:

realtime-chat-app.zip

STEP 2: Extract the ZIP File

1. Right-click on the ZIP file
2. Click Extract All
3. Choose a location (e.g., Desktop)

4. Click Extract

You will get a folder like:

realtime-chat-app/

STEP 3: Open Project in VS Code

1. Open Visual Studio Code
2. Click File → Open Folder
3. Select the extracted folder: realtime-chat-app
4. Click Open

STEP 4: Verify Project Structure

Inside the project folder, you should see:

realtime-chat-app/

├── client/

├── server/

├── README.md

└── README.pdf

STEP 5: Install Backend Dependencies

1. Open VS Code Terminal
2. Ctrl + `
3. Navigate to backend folder: cd server
3. Install dependencies: npm install

STEP 6: Start MongoDB (IMPORTANT)

Option A: MongoDB Local Service (Recommended)

Make sure MongoDB is running.

Check: mongosh

Option B: If MongoDB is not running

Start MongoDB service from:

- Windows Services → MongoDB Server → Start

STEP 7: Start Backend Server

Inside server folder: node server.js

You should see:

MongoDB connected

Server running on 5000

STEP 8: Install Frontend Dependencies

1. Open a **NEW terminal** in VS Code
2. Navigate to client folder: cd client
3. Install dependencies: npm install

STEP 9: Start Frontend Server

Inside client folder: npm run dev

You will see:

Local: http://localhost:5173

STEP 10: Open the Application

Open browser and go to:

http://localhost:5173

STEP 11: Register Users (IMPORTANT FOR CHAT)

1. Go to **Register page**
2. Create **User A**
3. Open **another browser / incognito**
4. Register **User B**

Two users are required for two-way chat

STEP 12: Login & Start Chat

1. Login as User A
2. Open Chat page
3. Login as User B (other browser)
4. Click each other's name
5. Send messages, images, files

STEP 14: Stop Servers

To stop servers safely:

- Backend terminal → Ctrl + C
- Frontend terminal → Ctrl + C