**🖥️ Multi-User Chat System – Frontend Documentation**

**Overview**

The frontend of the Multi-User Chat System is built with **React.js** and **Tailwind CSS**, offering a responsive, user-friendly interface. It supports **multi-user chat**, **media sharing**, **emoji input**, **real-time updates**, and **notifications**, ensuring a seamless communication experience across devices.

**UI Components and Their Purpose**

| **Component** | **Description** |
| --- | --- |
| **Chat Dashboard** | Displays all recent chat threads with users. Shows latest message and time. Lets users click to open a chat. |
| **User Search Bar** | Located at the top or sidebar. Allows users to search for other registered users by name/email. |
| **Chat Window** | Main area for sending/receiving messages. Displays message history, timestamps, sender info, and media. |
| **Message Input Box** | Text input field with emoji picker and file attachment options. Used to compose and send messages. |
| **Emoji Picker** | A popup interface integrated in the input area. Lets users select and insert emojis into messages. |
| **File Upload Button** | Allows uploading images, videos, or documents. Drag-and-drop supported in some views. |
| **Send Button** | Click to send the composed message. Supports keyboard shortcuts (e.g., Enter). |
| **Media Preview** | Shows thumbnails of uploaded files before sending. Click to remove or review. |
| **Notifications Panel / Toasts** | Displays real-time alerts (e.g., “New message from Alex”) without interrupting current view. |
| **Active Chat Highlight** | Visually indicates which chat is currently open. Often used in the sidebar or top nav. |

**Functional Flow**

1. **User Login & Dashboard Load**  
   After authentication, the user lands on the dashboard showing all previous chat sessions.
2. **Search & Initiate Chat**  
   Users can search by name or email to find other users. Clicking a user opens or creates a chat thread.
3. **Chat Interface**  
   Messages are sent and received instantly. Messages may contain text, emojis, images, or documents.
4. **Send Media**  
   Users can drag or browse to upload images/videos/documents. These are shown inline in the chat thread.
5. **Live Notifications**  
   Notifications appear for new messages, even if the chat is not currently open.

**Technologies Used**

* **React.js** – For component-based architecture and UI state handling.
* **Tailwind CSS** – For styling UI components with utility classes.
* **Socket.IO (Client)** – For real-time message delivery and typing indicators.
* **Emoji Picker (e.g., emoji-mart)** – Enables emoji selection in the chat.
* **React Dropzone / FilePond** – For uploading and previewing media files.

**Example Usage**

* A user logs in and sees a list of their current conversations.
* They search for a new colleague, open their profile, and begin chatting.
* They send a message, attach a video, and select a smiley emoji.
* The recipient sees the message in real time with a sound/visual notification.
* Later, the user receives a document from another chat with a preview and download option.