



COLLEGE CODE :- 9509

COLLEGE NAME :- Holy Cross Engineering College

DEPARTMENT :- CSE

STUDENT NM-ID:-

F56CFE83F6612DBB5FC42499263F8CA1

DATE :- 15/09/2025

Completed the project named as:

CHAT UI APPLICATION

SUBMITTED BY,

NAME :- Murugesh.K

MOBILE NO :-8300829750

1. Problem Statement

Traditional communication methods like email and SMS are slow, lack real-time interaction, and are not designed for quick, informal team collaboration. Users need a centralized, intuitive, and immediate way to communicate one-on-one and in groups. This leads to lost productivity, miscommunication, and a fragmented workflow.

2. Users & Stakeholders

User / Stakeholder	Role & Interests
End-User (Primary)	Team members, friends, or community members who need to send and receive messages in real-time. They value simplicity, speed, reliability, and a pleasant visual experience.
Group Administrator	A user who creates and manages group chats. They are responsible for adding/removing members and potentially moderating content. They need clear controls and moderation tools.
Product Manager	Wants to deliver a feature-rich, competitive product that meets user needs and achieves business goals (e.g., user engagement, retention).
UI/UX Designer	Focused on creating an intuitive, accessible, and aesthetically pleasing interface that ensures a smooth user journey and high usability.
Frontend Developer	Responsible for implementing the UI/UX designs using modern frameworks, ensuring the application is responsive and interacts correctly with the backend APIs.
Backend Developer	Responsible for building the server, WebSocket connections for real-time features, databases, and APIs that the UI depends on.
System Reliability Engineer (SRE)	Concerned with the application's scalability, uptime, and performance, especially under the load of multiple real-time connections.

3. User Stories

- **As a user,** I want to see a list of all my recent conversations so that I can quickly continue a past discussion.
- **As a user,** I want to send and receive text messages in real-time so that I can have a fluid conversation.
- **As a user,** I want to create a new group chat so that I can communicate with multiple people at once.
- **As a user,** I want to see when another user is typing a message so that I know a response is coming.
- **As a user,** I want to see the online/offline status of my contacts so that I know who is available to talk.

- **As a user**, I want to search for a specific message within a conversation so that I can quickly find important information.
- **As a group admin**, I want to add or remove participants from a group so that I can manage the conversation's audience.
- **As a user**, I want to have a visually clean and distraction-free interface so that I can focus on the conversation.

4. MVP Features (Minimum Viable Product)

The core features required for the first functional version of the UI:

1. Authentication UI:

- Login and Registration screens.

2. Dashboard / Conversations List:

- Sidebar showing all direct messages and group chats.
- Visual indicators for unread message counts.
- List item showing conversation name, last message preview, and timestamp.

3. Real-Time Chat Interface:

- Main message panel displaying the conversation history with sender name, avatar, and timestamp on each message.
- Message input area with a send button.
- Visual distinction between messages sent by the user and messages received from others.

4. Active Status Indicators:

- A green dot (or similar) next to usernames to show online status.
- A "typing..." indicator when a contact is composing a message.

5. Group Chat Management (Basic):

- UI Modal to create a new group by selecting from a list of contacts and naming the group.
- UI for group admins to add/remove members (stretch goal for MVP: could be Phase 2)
- .

6. Wireframes / API Endpoint List

Wireframes (Descriptive Overview):

• View 1: Login/Signup Screen

- Centered card with logo, input fields for email/username and password, and buttons for "Login" and "Sign Up".

• View 2: Main Application Dashboard

- **Left Sidebar (~25% width):** Contains a header with the user's avatar and a "New Chat/Group" button. Below is a scrollable list of conversations.

○ Main Panel (~75% width):

- *Default State:* If no chat is selected, displays a welcome message.
- *Active Chat State:* Header with conversation name and participant status. Central area is a scrollable message history. Fixed footer contains the message input box and send button.

API Endpoint List (What the UI will consume):

The UI will interact with a backend via these typical RESTful API endpoints and a WebSocket connection.

Method	Endpoint	Purpose	UI Action
POST	/auth/login	User login	Submit login form
POST	/auth/register	User registration	Submit signup form
GET	/conversations	Get user's conversation list	Populate the left sidebar
GET	/conversations/{id}/messages	Get messages for a specific chat	Load message history when a chat is selected
POST	/conversations/{id}/messages	Send a new message to a chat	User hits the "send" button
POST	/conversations/group	Create a new group chat	Submit "create group" modal form
POST	/conversations/direct	Start a new direct message	Click "New Chat" on a user
PATCH	/conversations/{id}/read	Mark messages as read	When a user views a chat
WS	ws://	WebSocket Connection	Real-time: receiving new messages, typing indicators, presence updates.

6. Acceptance Criteria

- **Authentication:**

- Given correct credentials, the user is redirected to the main dashboard upon clicking "Login".
- Given incorrect credentials, a user-friendly error message is displayed.

- **Conversations List:**

- The left sidebar loads and displays all conversations the user is a part of upon page load.
- The most recent conversation is at the top of the list.
- Unread messages are clearly indicated with a numeric badge.

- **Message Display:**

- When a conversation is clicked, the main panel loads its message history, sorted with the newest at the bottom.
- Messages from the current user are aligned to the right and styled differently (e.g., different color).
- Messages from others are aligned to the left.
- Each message displays a timestamp in a sensible format (e.g., "HH:MM").

- **Sending Messages:**

- After typing text and clicking "Send", the message immediately appears in the message history.
- The input field is cleared after sending.

- **Real-Time Features:**

- A message sent by another user appears in the current user's active chat instantly without requiring a page refresh.
- The "typing..." indicator appears in the chat header when another participant is typing and disappears when they stop or send.
- Online status (green dot) is shown for contacts who are currently active.