USER GUIDE - Secure Chat

(Project 1- Group 23 - CPSC 455-02 14041)

Sana Mansoori (CWID: 883033920, sana.mansoori@csu.fullerton.edu)

Jaytee Okonkwo (CWID: 890207855) jokon@csu.fullerton.edu)

I. INTRODUCTION

SecureChat is a real-time chat application built with **FastAPI** and **WebSockets**. It lets multiple users connect, send **encrypted** messages & files instantly, and handles presence and reconnection gracefully.

Since our last release, SecureChat has been enhanced with:

- **Server Deployment & DB Hosting**
- **Secure File Sharing (Cloud Friendly)**
- **End-to-End Encryption** for both DMs and group chats (messages & files)
- **Presence Detection** (online/offline/typing)
- **24/7 Uptime Monitoring**

II. TECHNOLOGIES USED

- FastAPI– Handles the web interface and user session management, API endpoints, and server-side handling.
- WebSockets – Enables real-time communication between users.
- HTML/CSS/Javascript - Acts as the WebSocket user-friendly web client interface.
- Flask-SQLAlchemy – Manages user database and authentication using SQLite.
- Flask-Bcrypt – Securely hashes user passwords.
- SSL/TLS Encryption – Secures WebSocket communication.
- AES-GCM - Encrypts file data and direct chat messages
- ECDH- Derives a shared key exchange for E2E in one-to-one chats
- Quill Editor & Emoji Picker provides rich text formatting and emoji support.
- VirusTotal API – malware scanning
- Firebase – encrypted file storage
- CockroachDB - serverless database cluster hosting using PostgreSQL

III. HOSTED WEBSITE

The application is available at: 🔗 https://securechat-oe69.onrender.com/home
The WebSocket server runs at: 🔗 wss://securechat-oe69.onrender.com/ws

3.1 PREREQUISITES

- Web browser (Google Chrome, Firefox, etc.)

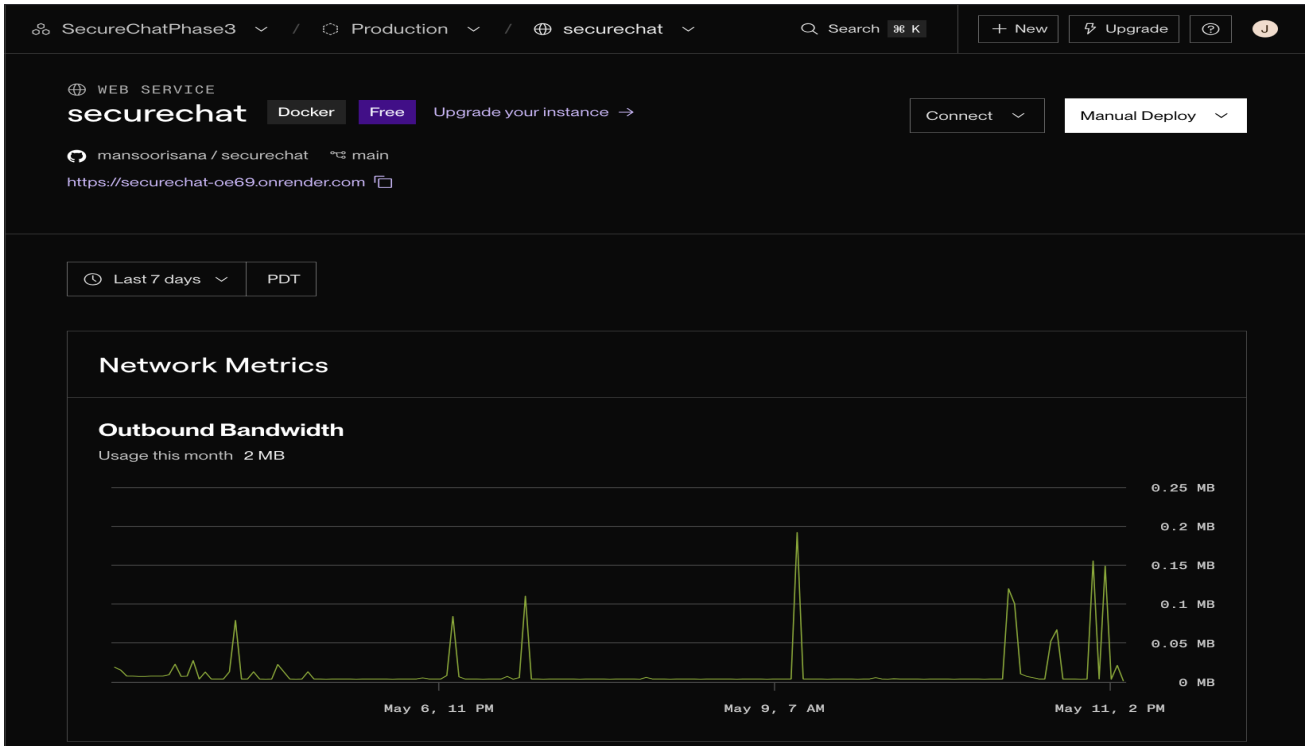
3.2 GITHUB REPOSITORY

You can access the GitHub repository https://github.com/mansoorisana/securechat
Or even clone it using below commands:

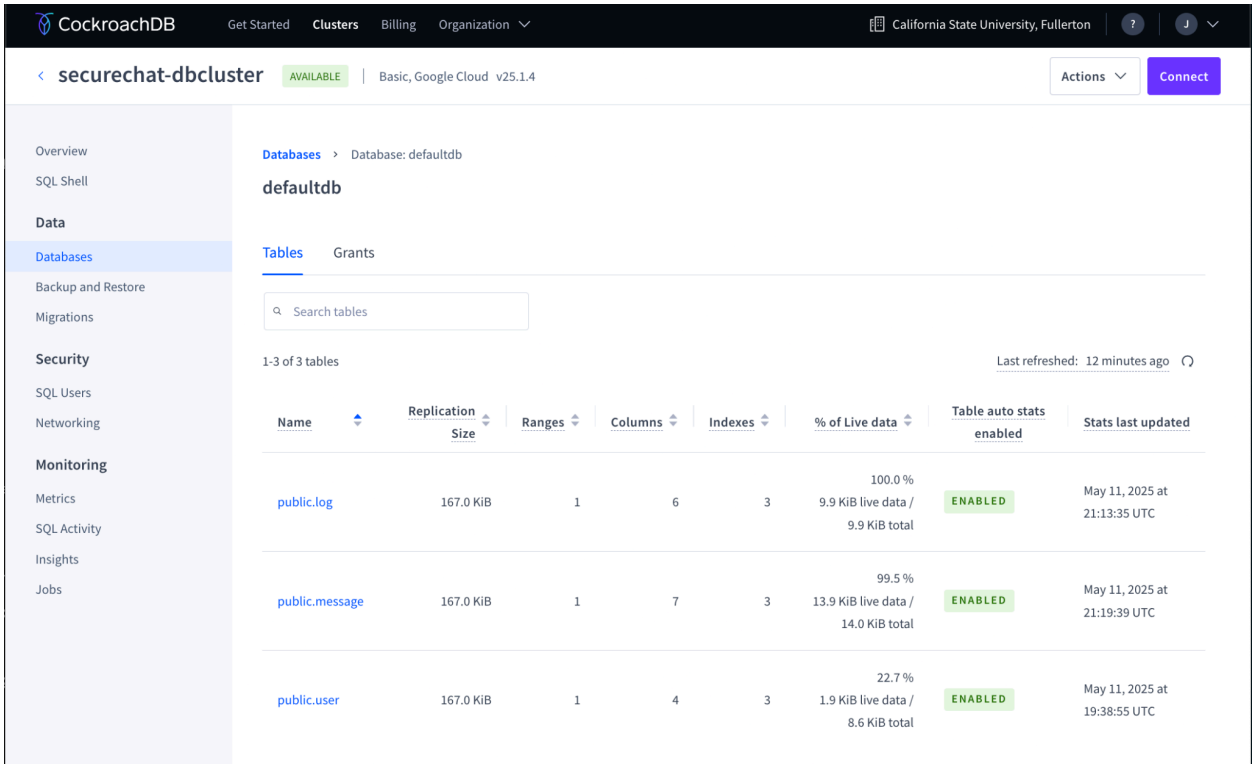
git clone https://github.com/mansoorisana/securechat.git
cd yourrepo

Refer below screenshot for server setup & other services used:

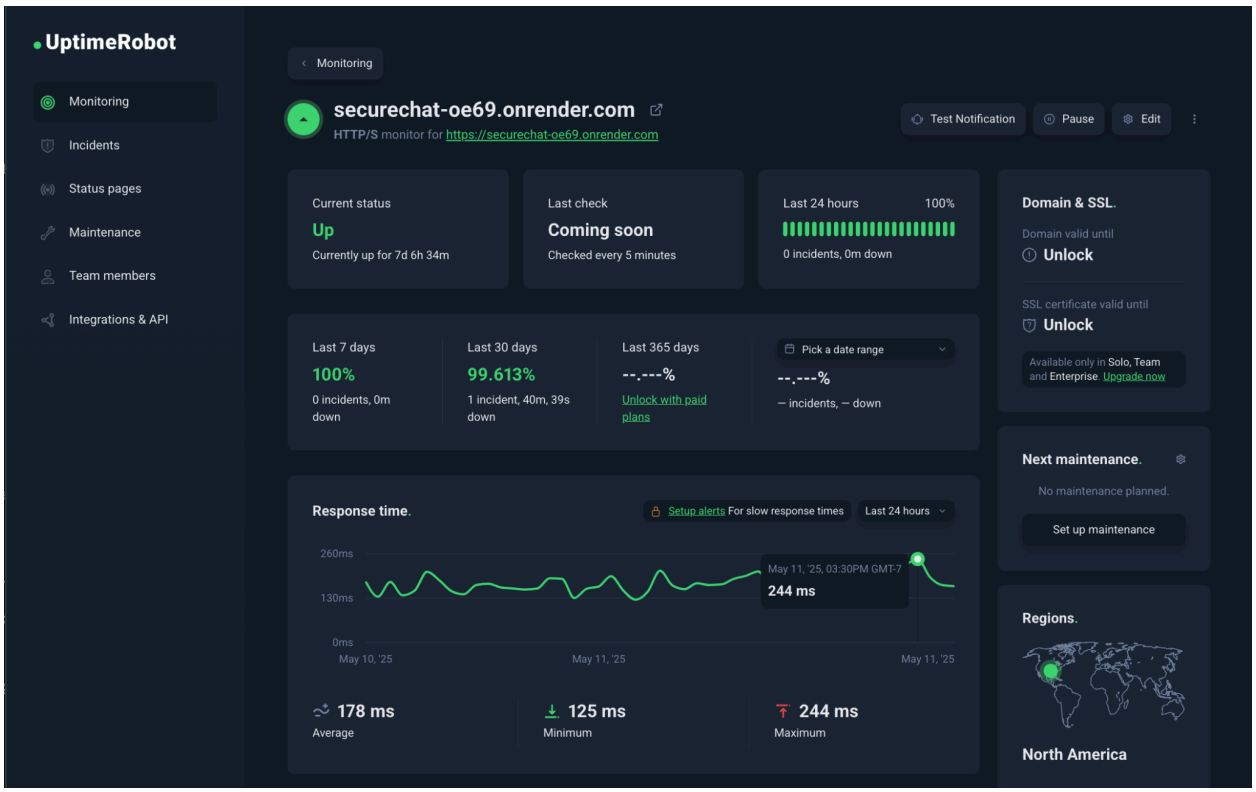
***Render.io*:**



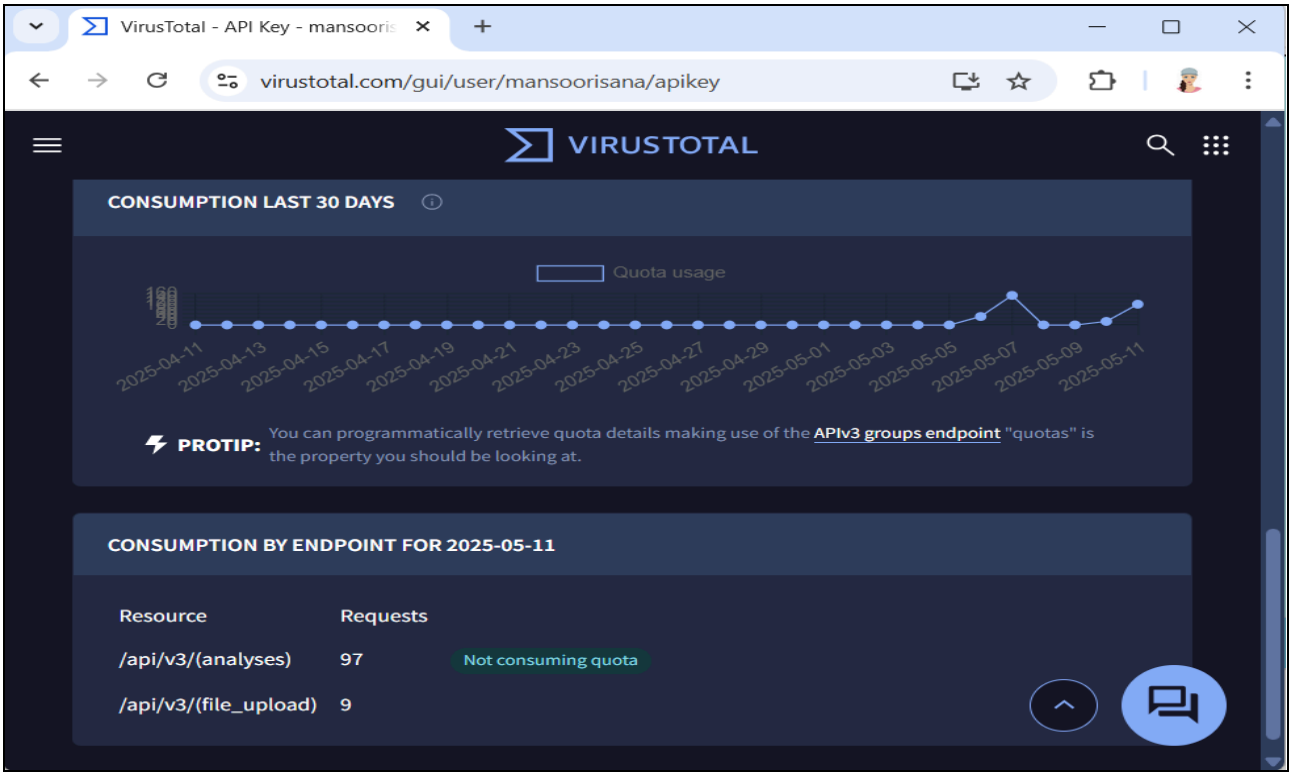
CockroachDB:



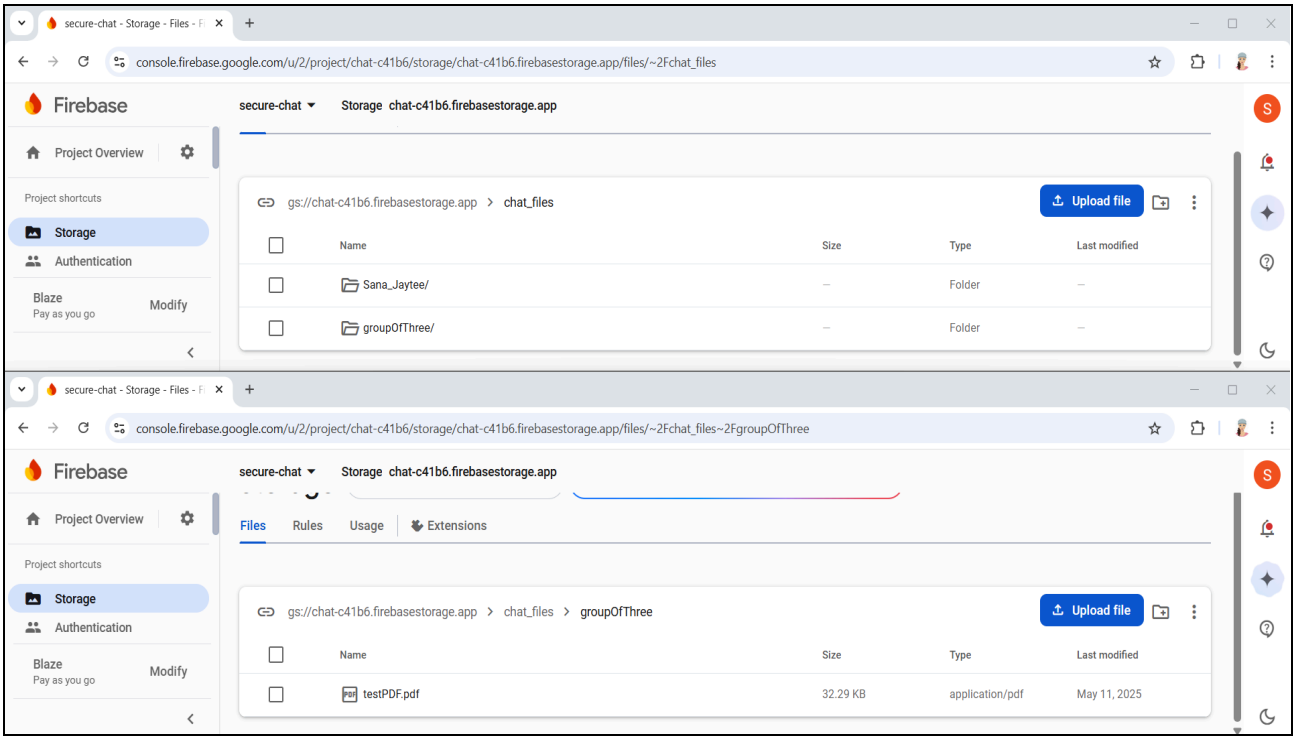
UptimeRobot:



VirusTotal:



Firebase Storage:



IV. USING THE CHAT APPLICATION

4.1 ACCESSING THE CHAT ROOM

1. Open https://securechat-oe69.onrender.com/home in your browser.
2. Signup with username & password.
3. You will be prompted to login after successful signup & click **Login**
4. You will be redirected to the chat room https://securechat-oe69.onrender.com/chat

4.2 USER AUTHENTICATION

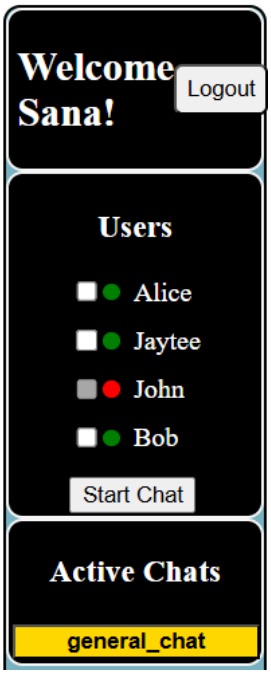
1. Brute-force protection is enforced on login attempts with 5 attempts in 5 minutes per IP address
2. Client authorization and authentication happen on the server side, using bcrypt to hash all passwords stored in the database.

4.3 CHAT PAGE GUI - CREATING CHAT

User List:
The left panel displays available users. The green and red dot shows user's presence status.
Direct Chats:
Select only one checkbox corresponding to the username for encrypted private chat.

Group Chats:
Select multiple checkboxes for group chat. Then, enter a name for the group. Start messaging for encrypted group chat.

General Chat Room:
A public chat room where all users can chat and read messages



4.4 CHAT PAGE GUI - CHAT SELECTION & WINDOW

1. Left side panel displays **Active Chats.**
2. Click on any of the listed chats to send messages on that particular chat.
3. Selected chat will open on the right side chat window with chat title.
4. Default active chat is the **general_chat** containing all the active users.
5. All messages display a timestamp next to the message, for tracking purposes.

4.5 SENDING MESSAGES

- Type a message in the input box and press **Send**.
- Click on the **emoji** icon to display emoji picker.
- Format text by using **bold, italics, underline, strikethrough, headings, superscript/subscript, bullets/numbering list, change font color & text highlight** icons provided in the chat box.
- All users can see ***"<username> typing…"*** in the currently open chat window if someone is typing in that chat.
- Messages will appear instantly in the current active chat room.
- Direct & Group Messages between users are end-to-end encrypted.

4.6 FILE SHARING

- File Uploads:
  Simply press the file upload button, and you will be prompted to select a file to

upload. A pop-up will appear, ***"Please wait at least 20 seconds for file scan & upload!"*** as the file is first scanned for malware. If the scan is positive, file uploads are encrypted before being sent to cloud storage.

- File Downloads:
  Press the file download button and select a file from the modal file to download. Only encrypted files sent in chat are downloadable and decrypted using a decryption key on the client side.

4.7 CONNECTION HANDLING

- INITIAL CONNECTION:
  - The server listens on wss://securechat-oe69.onrender.com/ws with SSL.
  - When a client connects, it sends a username as the first message.
  - The server verifies the username against the database before allowing communication.

- DISCONNECTION:
  - Click **Logout** to exit.
  - Your session will be cleared.

- RECONNECTION:
  - Server uses a PING/PONG mechanism with ping_interval = 10s and ping_timeout = 5s to detect broken connections.
  - If there is no response within the timeout, the client is considered disconnected.
  - Client attempts reconnection after 3 seconds if an unexpected interruption occurs.

4.8 RATE LIMITING

- Users can send up to 5 messages in 10 seconds.
- If a user exceeds this limit:
  - They receive a warning message:
    "RATE LIMIT EXCEEDED. PLEASE WAIT FOR 10 SECONDS."
  - Messages sent during the mute period are ignored.

V. CONCLUSION

SecureChat now includes:

- Public web hosting on Render.
- 24/7 Uptime monitoring using UptimeRobot.
- Message & File Encryption:
    - One-to-one direct messages/files are end-to-end encrypted using ECDH key exchange and AES-GCM, ensuring user privacy.
    - Group messages/files are end-to-end encrypted using the encrypted group key shared by the group creator for each chat member.
- Malware scan before file upload using VirusTotal.
- End-to-end encrypted per chat file hosting on Firebase.
- Presence Detection with “online”, “offline” and “typing…” indicators.