

CHIT-CHAT: Full Stack Chatting App

System Design Document

Overview

Chit-Chat is a robust Full Stack Chatting App designed to provide users with a seamless and secure communication platform. The app uses React JS for the client-side interface, Node JS and Express JS for the server-side development, and MongoDB as the backend database. Real-time communication is facilitated through Socket.io, ensuring instant messaging capabilities. User details are stored in an encrypted format in MongoDB to prioritize data security.

Tech Stack

- Client:
 - React JS:
 - React JS is chosen for its efficient and modular component-based architecture. It enables the development of dynamic and responsive user interfaces, providing an excellent user experience.
- Server:
 - Node JS:
 - Node JS is used for server-side development, offering a non-blocking, event-driven architecture that makes it well-suited for handling concurrent connections.
 - Express JS:

- Express JS is utilized as the web application framework for Node JS, simplifying the creation of robust APIs and handling HTTP requests.
- **Real-Time Communication:**
 - Socket.io:
 - Socket.io is employed for real-time bidirectional communication between clients and the server. It ensures instant messaging features, including typing indicators and real-time updates.
- **Database:**
 - MongoDB:
 - MongoDB is chosen as the NoSQL database for its flexibility and scalability. User details are stored in an encrypted format to enhance data security.

Features

Authentication:

- Users are required to authenticate themselves securely before gaining access to the chat features.

Real-Time Chatting:

- Utilizes Socket.io to enable real-time communication, ensuring that users can exchange messages instantly.

Search Users:

- Enables users to search for and connect with other users, streamlining the process of finding and adding friends.

Create Group Chats:

- Users can create and participate in group chats, fostering collaboration and communication among multiple participants.

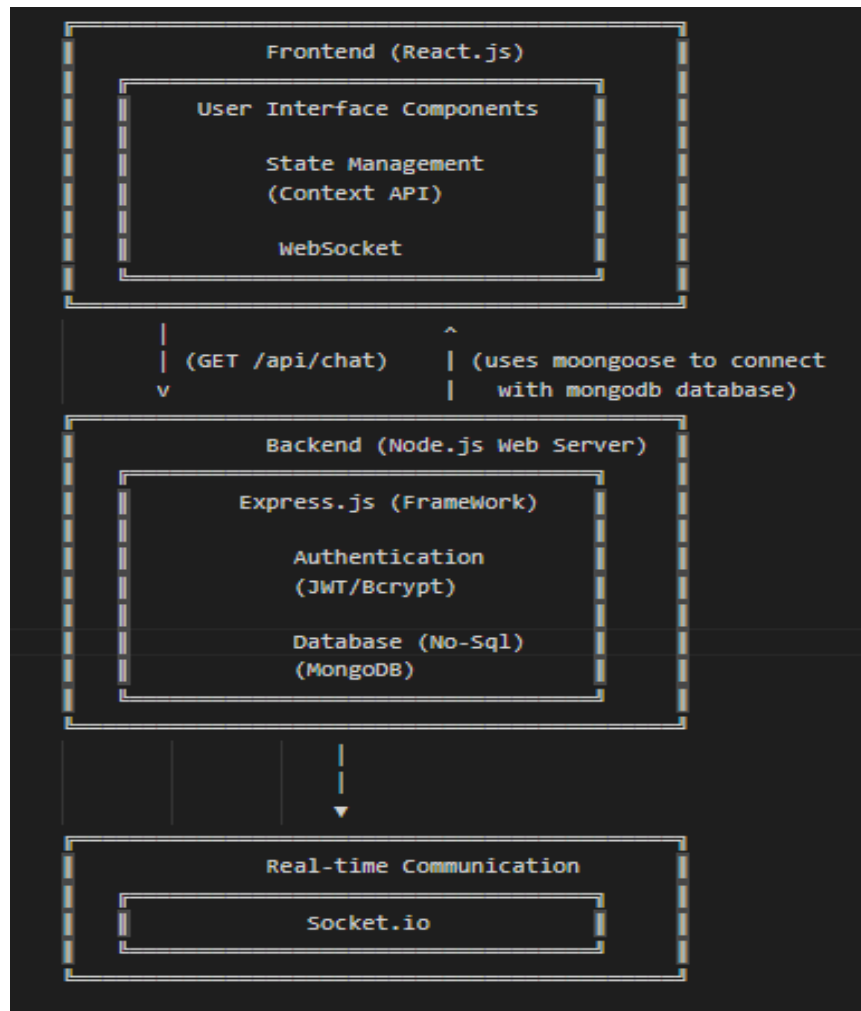
Add or Remove Users from Group:

- Group administrators have the capability to manage group membership by adding or removing users, enhancing control over group dynamics.

View Other User Profile:

- Users can view profiles of others, providing additional context and personalization to the chat experience.

System Design:



Run Locally

To run the Chit-Chat webapp locally on your machine, follow these steps:

Clone the project:

- **git clone** <https://github.com/AnjaliPatil4/Chit-Chat.git>

Navigate to the project directory:

- **cd Chit-Chat**

Install dependencies:

- **npm install**
- **cd frontend**

- **npm install**

Start the server:

Run the command to start the server.

- **npm start**

Start the Client:

Open a new terminal or command prompt, navigate to the frontend folder, and run:

- **Cd frontend**
- **npm start**

Following these steps will set up a local instance of Chit-Chat, allowing you to explore its features and functionalities in your development environment. Make any necessary adjustments based on your specific setup or preferences.

Dependencies and Libraries

- Socket.io:
 - Utilized for real-time bidirectional communication, crucial for implementing instant messaging features.
- React JS:
 - Chosen for its efficient and modular component-based architecture, facilitating the creation of dynamic user interfaces.
- Node JS and Express JS:
 - Selected for server-side development, offering a robust and scalable backend with a focus on building RESTful APIs.
- MongoDB:
 - Chosen as the database for its flexibility and scalability, storing user details in an encrypted format to ensure data security.

Why These Technologies?

- React JS:
 - Provides an efficient and modular component-based architecture for building dynamic and responsive user interfaces.
- Node JS and Express JS:

- Enables fast and scalable server-side development with a focus on building RESTful APIs.
- Socket.io:
 - Facilitates real-time bidirectional communication, crucial for implementing instant messaging features.
- MongoDB:
 - NoSQL database offers flexibility and scalability, suitable for handling dynamic and evolving chat data.
- Authentication: JWT (JSON Web Tokens)
 - JWTs provide a secure way to authenticate users by generating a token that includes user information.
- Password Hashing: Bcrypt
 - Bcrypt is a secure and widely used password-hashing algorithm, ensuring the safety of user credentials.

Screenshot of Interface

Sign Up

Chit-Chat

Login

Sign Up

Name *

Email Address *

Password *

Show

Confirm Password *

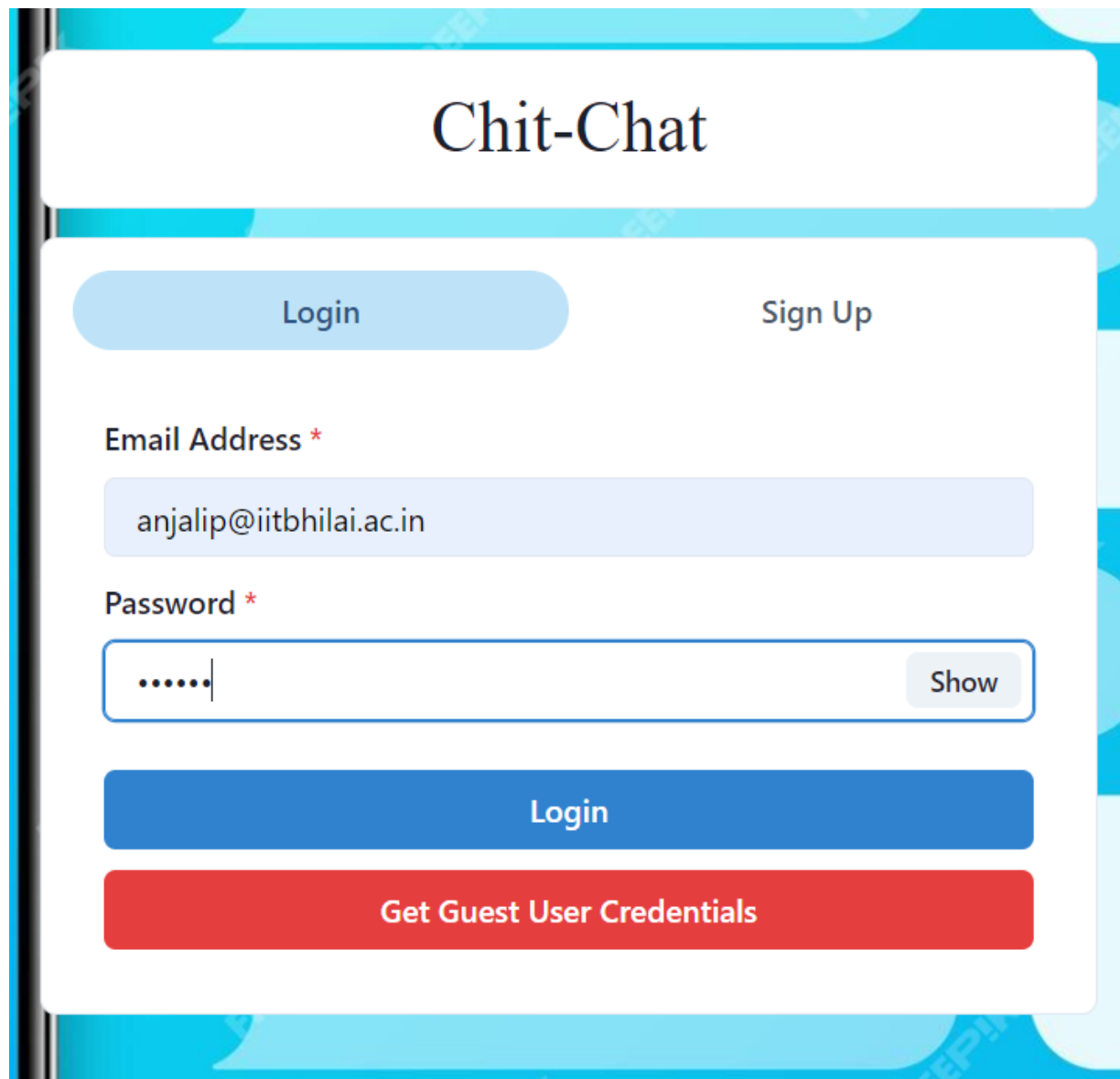
Show

Upload your Picture

 No file chosen

Sign Up

Login

The image shows a web form titled "Chit-Chat" with a light blue background. At the top, there are two buttons: "Login" in a light blue rounded rectangle and "Sign Up" in a white rounded rectangle. Below these, the form asks for an "Email Address *" with a text input field containing "anjalip@iitbhilai.ac.in". The next field is "Password *" with a text input field showing six dots and a "Show" button to its right. At the bottom, there are two large buttons: a blue "Login" button and a red "Get Guest User Credentials" button.

Chit-Chat

Login Sign Up

Email Address *

anjalip@iitbhilai.ac.in

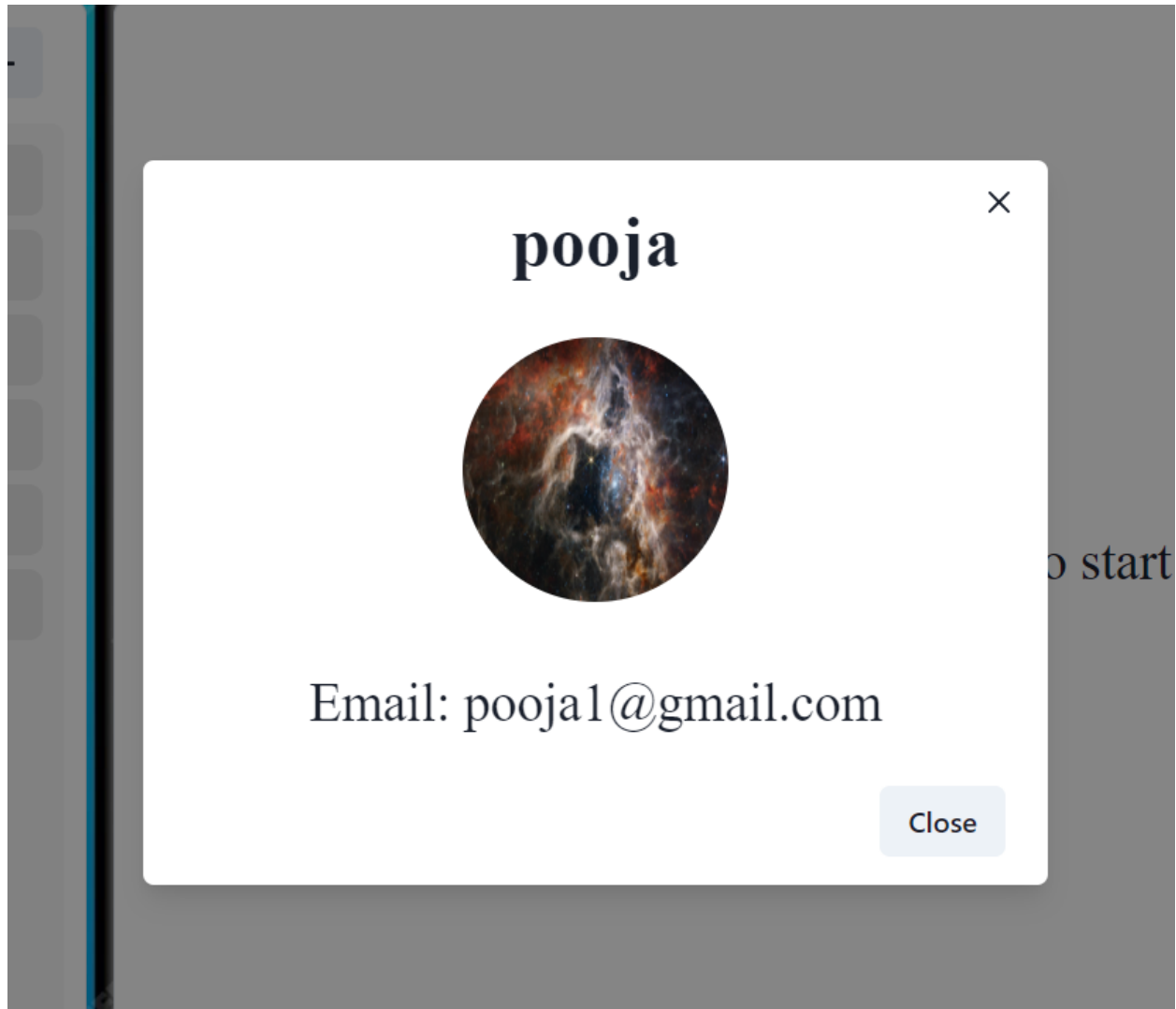
Password *

..... Show

Login

Get Guest User Credentials

You can view your Profile



Search User

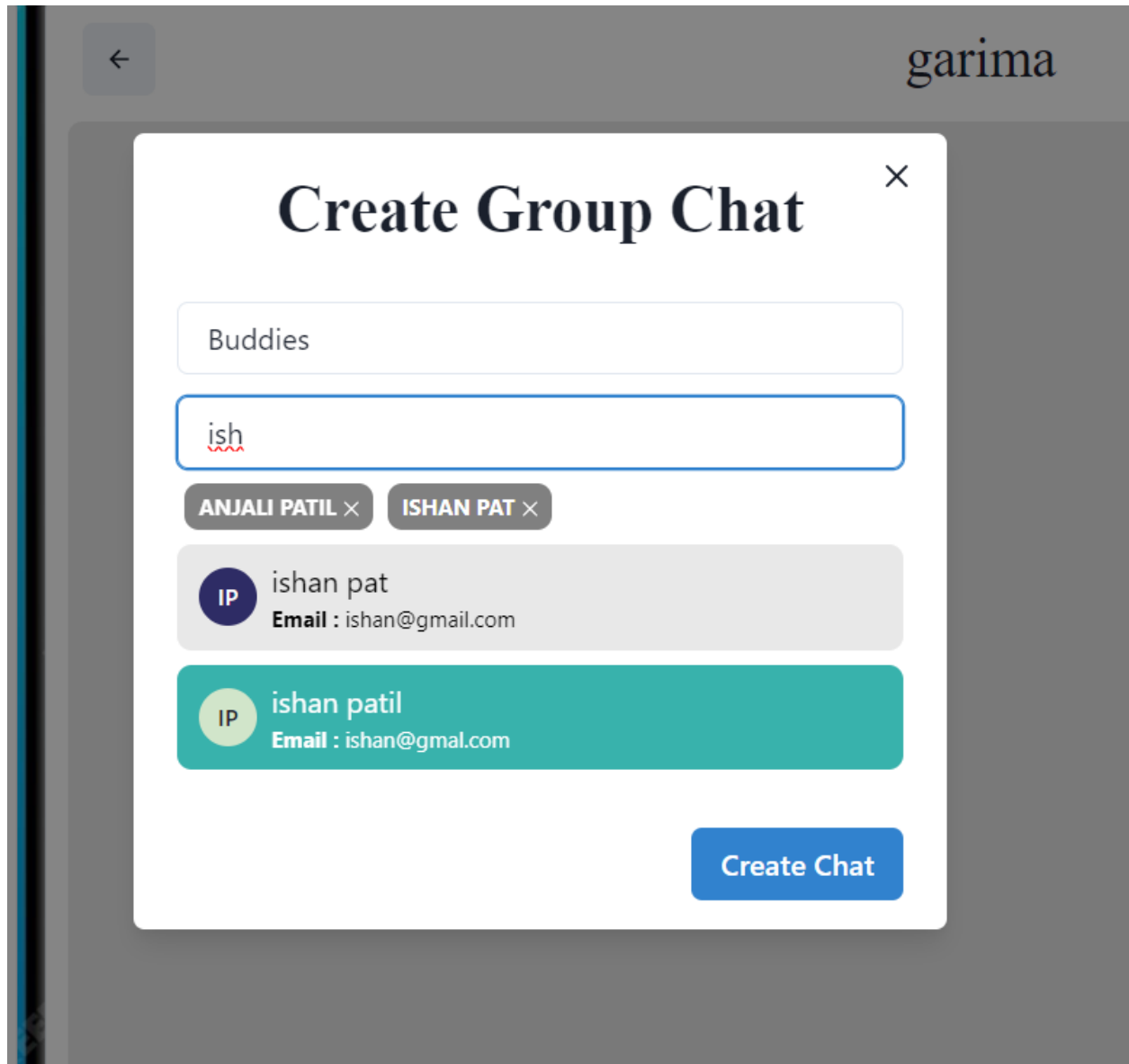
Search Users



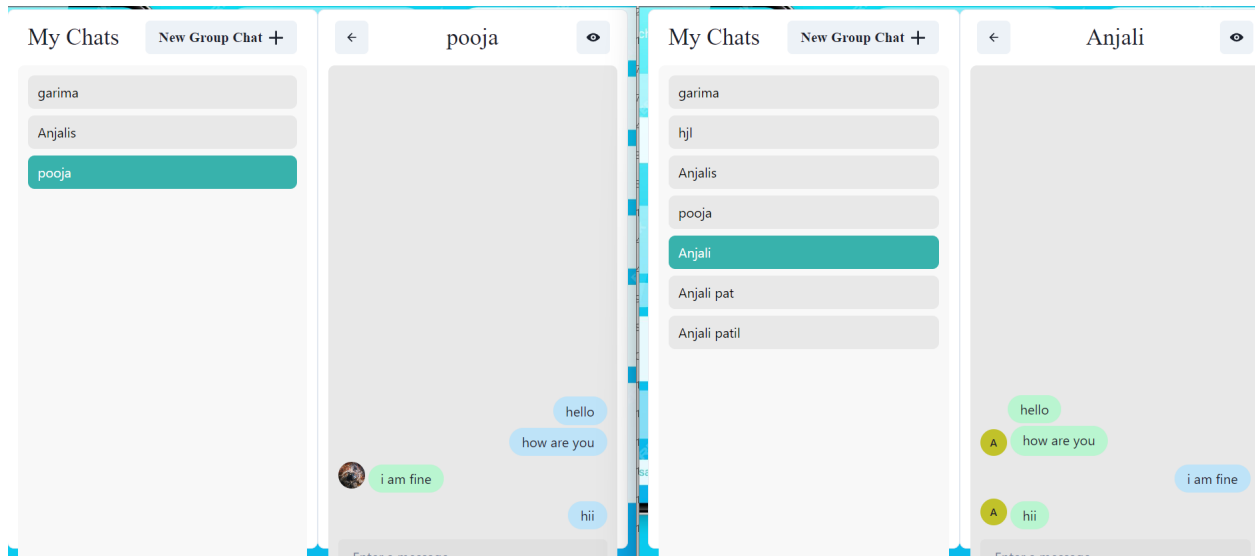
garima

Email : garima@gmail.com

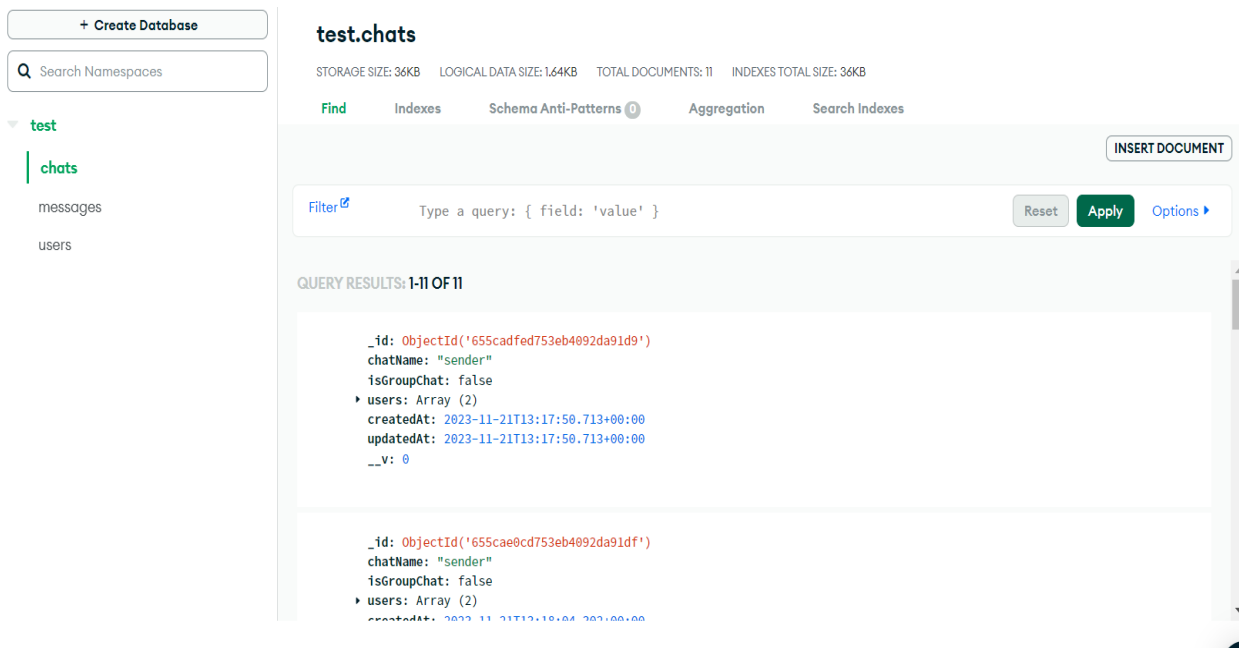
Create Group Chat(Admin can Add or remove users from group)



Real Time Communication



Database(Mongodb)



Conclusion

Chit-Chat Is a feature-rich chat application that prioritizes real-time communication and user privacy. The chosen technologies and architecture ensure a smooth and responsive user experience, making it an ideal solution for modern messaging needs.