**Chat Application**

This chat application allows users to communicate in real time. It supports both private one-to-one chats and group chats. The application is built using Flask, Flask-Socket-IO, Jinja2, and MongoDB. Users can join a group or start a one-on-one chat and exchange messages in real time.

**Key Features-**

* **Real-Time Messaging:** Users can send and receive messages instantly using Web-Sockets (Socket.IO).
* **Group Chats:** Users can create groups and chat with multiple users in a single room.
* **Private Chats:** One-to-one chat functionality is available based on user IDs or email.
* **Persistent Chat History:** All messages are stored in MongoDB and can be retrieved when the user returns.
* **Multiple Rooms:** Dynamic rooms are created for each private chat and group chat.

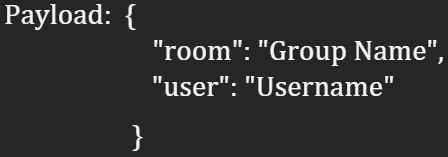
**Backend: Flask Application-**

* **Flask Routes and Functions**

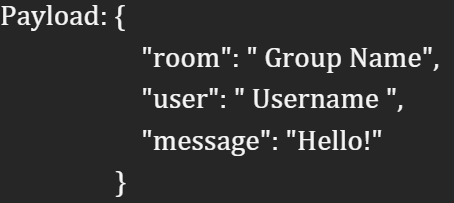
1. **/signup\_page** **API –** Displays the sign-up page for new users.
2. **/signin\_page** **API –** Displays the login page for existing users.
3. **/signup API –** Handles the user registration process.
4. **/signin** **task API –** Handles user login and session creation.
5. **/logout** **API-** Logs out the user and clears their session.
6. **/active\_users/<access\_email>/<access\_email\_name> API-** Shows all users who signed up for it.
7. **/create\_join\_group/<access\_email>/<access\_email\_name> API-** Allows users to create a new group by providing a group name.
8. **/created\_groups/<access\_email>/<access\_email\_name> API-** Shows the groups owned by the user and the groups they have joined.
9. **/single\_chat\_page/<sender\_email>/<receiver\_email>/<receiver\_name>/<access\_email\_name> API-** Opens a private chat between two users.
10. **/createGroup/<access\_email>/<access\_email\_name> API-** Create a new group for group chat.
11. **/joinGroup/<access\_email>/<access\_email\_name>** **API-** Join a group using access key.
12. **/group\_chat\_page/<sender\_email>/<access\_email\_name>/<groupName>/<groupAccessKey>** **API -** Opens the chat room for a specific group.

* **Socket.IO Events**

1. **Join-** Allows a user to join a chat room.

****

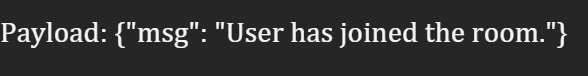
1. **Send\_message-** Sends a message to a room.



1. **New\_message-** Broadcasts a new message to all users in a room.

****

1. **Status-** Broadcasts a status update.(e.g, available sign, green dot, user joined room)



**Frontend: Jinja2 templates-**

* **Templates-**

1. **Authentication -** index.html, header.html, signup.html, signin.html
2. **User landing Templates-** dashboard.html, NavBar.html
3. **activeUser.html-** Shows all users who signed up for application.
4. **createJoinGroup.html-** Allows users to create a new group by providing a group name.
5. **createdGroups.html-** Shows the groups owned by the user and the groups they have joined.
6. **chat.html-** private chat between two users.
7. **groupChat.html-** group chat for a specific group.

**Conclusion-** The chat application is a real-time communication platform built with Flask, Flask-Socket-IO, Jinja2, and MongoDB. It supports private and group chats with features like online status indicators (green dot for active users), session management for persistent logins, and dynamic group member visibility. Users can see group members, and non-participants are excluded from group views. Bootstrap 4 is integrated for a responsive, mobile-friendly design with optimized layouts. The app ensures seamless real-time updates and a clean, user-friendly interface.