**Real-Time Chat Application with React, Flask, Firebase, and Google Authentication**

Introduction:

We propose to build a real-time chat application using React for the frontend, Flask for the backend, Firebase as the database, and Google Authentication for user authentication. The application will allow users to create accounts, join chat rooms, and exchange messages in real-time. Users will be able to login using their Google accounts, and their authentication will be handled by Firebase.

Features:

**User Authentication**: Users will be able to create accounts and login using their Google accounts for authentication. Firebase will handle the user authentication process, including secure storage of user credentials.

**Profile Management**: Users will be able to manage their profile information, including their profile picture and display name. Users will be able to update their personal information and view their activity history.

**Chat Rooms**: Users will be able to join chat rooms based on different channels, such as news, sports, food, dogs, and announcements. They will be able to exchange messages with other users in real-time within the chat rooms.

**Announcement Feature:** Admins of the application will be able to create announcements and broadcast them to all users. Users will be able to view announcements and see the time and date they were posted.

**Logout Functionality**: Users will be able to logout from their accounts, which will securely invalidate their session and protect their account information.

Database:

Firebase, a NoSQL cloud-based database, will be used to store the data for the application, including user profiles, chat messages, and announcements. Firebase provides real-time synchronization and scalable storage, making it suitable for a real-time chat application.

Backend:

Flask, a micro web framework, will be used for the backend of the application. Flask will handle the user authentication process with Firebase, as well as handle the creation and management of chat rooms and announcements. Flask will also establish an instance to enable real-time communication between clients within the chat rooms.

Frontend:

React, a popular JavaScript library for building user interfaces, will be used for the frontend of the application. React will handle the rendering of the chat room interface, user profile management, and login/logout functionality. React components will communicate with the Flask backend to send and receive data from Firebase.

Responsibilities:

Suraj Mandal will be responsible for developing the frontend using React and integrating it with the Flask backend.

Vedavyas Venkata Narasimha Vedavyas Muppavarapu will be responsible for developing the backend using Flask, integrating with Firebase for user authentication and database management.

Conclusion:

This project aims to create a real-time chat application using React, Flask, Firebase, and Google Authentication. The application will allow users to create accounts, join chat rooms, and exchange messages in real-time. The use of Firebase for the database and Google Authentication for user authentication will provide a secure and scalable solution. The combination of React and Flask will result in a modern and responsive web application with real-time chat functionality.