Chat App

## Team Members

Greg Barré, Alina Gotcherian, Ali Nehme

## Overview

Our project is a messaging app/forum similar to Discord, built with a React.js frontend and Node.js backend. A user will be able to register and log in to the app and can update their profile information such as writing a bio or changing their chat avatar. While logged in, a user will be able to interact with other users within the same channel.

A channel is a conversation thread between two or more users where one channel member has the status of admin and can add or remove other users. In channels of only two users, the initiator of the conversation becomes the admin. Messages are displayed chronologically as they are posted by users and remain visible only to other users in the same channel.

Users within the same channel can react to each other’s individual messages, in the form of a “like”. Likes are togglable; if user clicks like on a message that they have previously liked, the original like will be removed.

The channel admin can “pin” any messaged deemed particularly important to make it appear at the top of the message list and remain highly visible to users in the channel.

A search function will allow users to filter messages based on keywords and provide quick access to relevant messages.

• **Technologies used**: React.js (frontend), Node.js (backend), MySQL (database).

• **Special features**: Ability to “pin” a message, reacting to (liking) messages, CRUD of users by the channel admin, search function to filter messages.

• **Additional Libraries:** Server: Express, CORS, MySQL2, Nodemon, Sequelize, BCrypt, JWT – Client: Axios, React-Router-DOM, Formik, Yup, Bootstrap. Socket.IO

• **Challenging features:** Video Calling with (webRTC), sharing of media (files, pictures, video, etc.).

## Routes

/Login

/Register

/ (index)

/Admin

/Admin/Users

/Channel/

/Channel/Admin

/Message

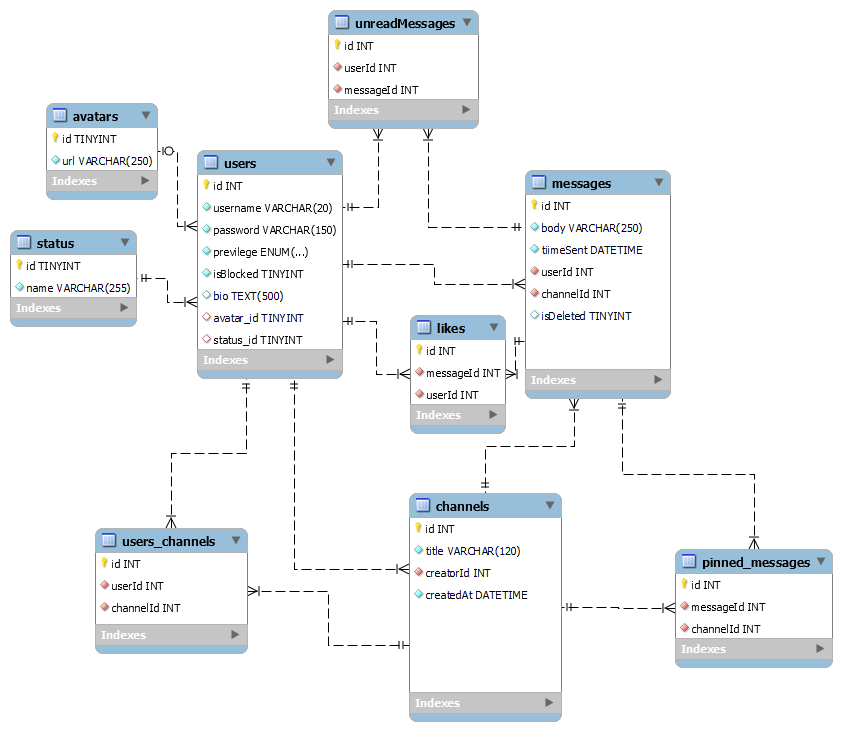
/User

/User/Profile

/User/Channels

/User/Messages

## Database Schema



## Snapshots

After Registration/Login, and aside from admin features & personal profile editing, most of the components of the app will be rendered on a single page that will be divided to show channels on one side, the conversation in the middle, and a text box to chat at the bottom.

**Example 1 - *Slack***

Graphical user interface, text, application, chat or text message

Description automatically generatedInspiration from **Slack** for the chat **search** feature, **channels**, **chats** and **message** **box**, **profile** **avatar**, message **reactions** (ours will only include Likes)

**Example 2 - *Discord***

Graphical user interface, text, application, chat or text message

Description automatically generated

Inspiration from **Discord** for the chat **search** feature, **channels**, **chats** and **message** **box**, **profile** **avatar**, message **reactions** (ours will only include Likes)

## Questions

* Are there any libraries or tools you would recommend helping us achieve our project idea?
* What are the most key features you think we should focus on?
* Do you foresee us running into any problems with this project idea?
* Is there an effective way to test this project?