Super Team
667 Final Project
Team members:
Jinghan Cao
Jordan Carlson
Thomas Davey
Cody Xu
Jiawei Xu
Diana Yu Yu

Instant message application

I. Introduction

Throughout this class, we got exposure to different technologies and frameworks that have helped us to gain the necessary knowledge on how to build a web application. Given the theme of the project, after finishing homework 3, our team decided to design a project based on instant messaging where users are able to view real-time updates of messages, create a user account, search for a specific user, create a chat between more than two people.

II. Technologies

  A. **JavaScript** as the coding language
  B. **Visual Studio Code** as the IDE for this project
  C. **React and Redux** which are JavaScript libraries used mainly for the user interface of this project and managing the web app components
  D. **Redis** in order to store users username and passwords on the browser via cookies
  E. **MongoDB** a simple and useful database for storing the user's information
  F. **Websocket** used for real-time updates of the app
  G. **Docker:** Tool used for deploying our application by using containers
  H. **Kafka:** Framework used for storing and analyzing the streaming data, in this case, the conveyor of the project

III. How to run the project locally

Before running any files, it is important that we run locally MongoDB and Redis in a local environment.

  A. **npm install:** this command will install all the dependency packages
  B. **node gateway.js:** The file that deals with all the ports of the backend

C. **node messenger.js:** You need to run this command so that this service can be linked to MongoDB to communicate message data.
D. **node user:** To keep this service on to ensure user login and register functions.
E. **node websocket.js**: This file is in charge of the real-time update between the messages and the different users
F. **npm run start:** Final step to initialize the react application

IV. **Address of the website**

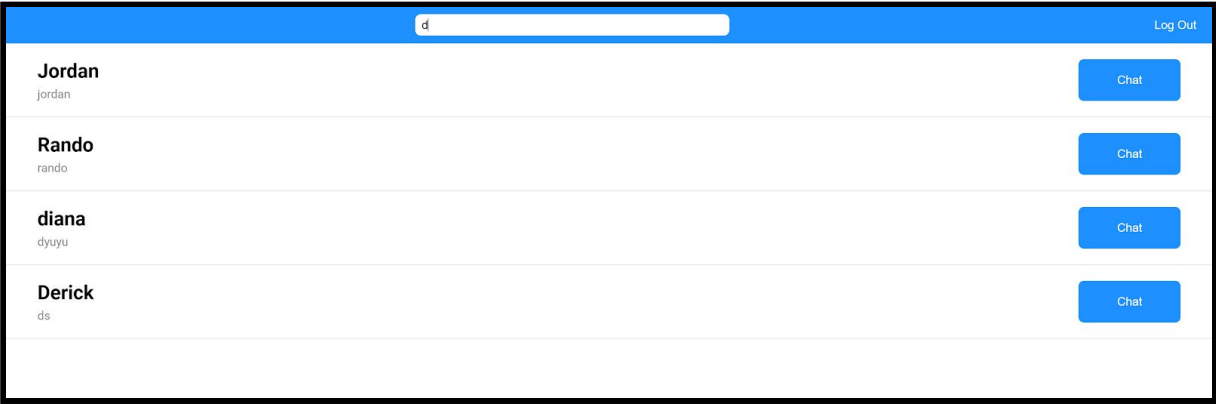A. **http://18.188.240.145:4000/**
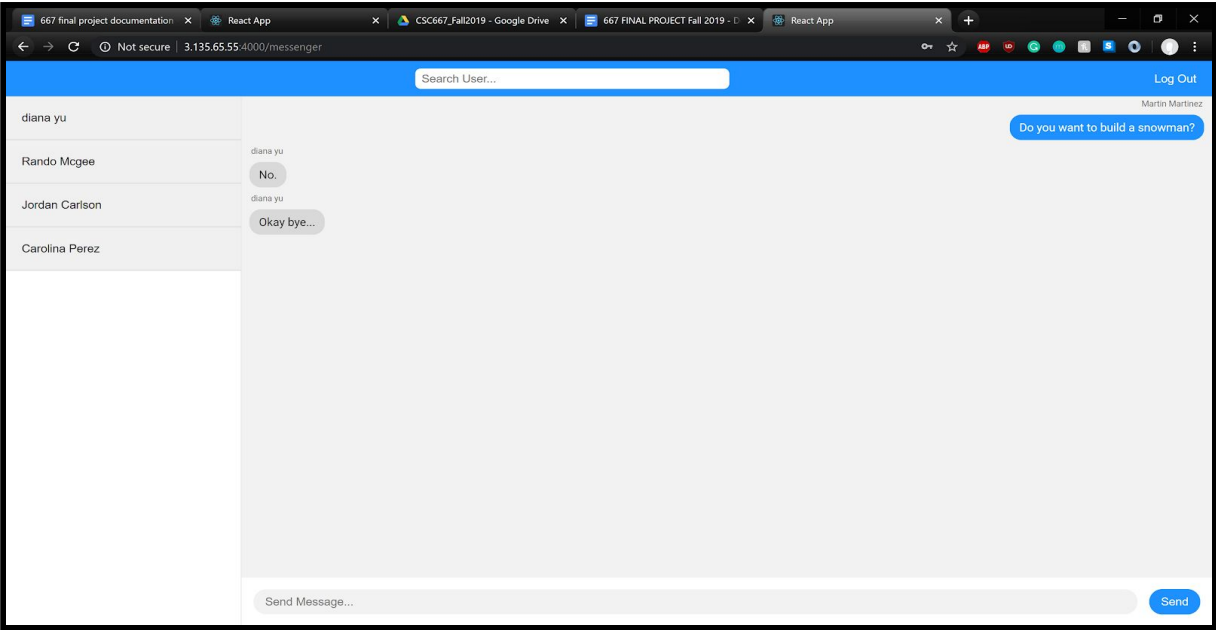
V. **UI mock-up**

Simple Login and Register Page
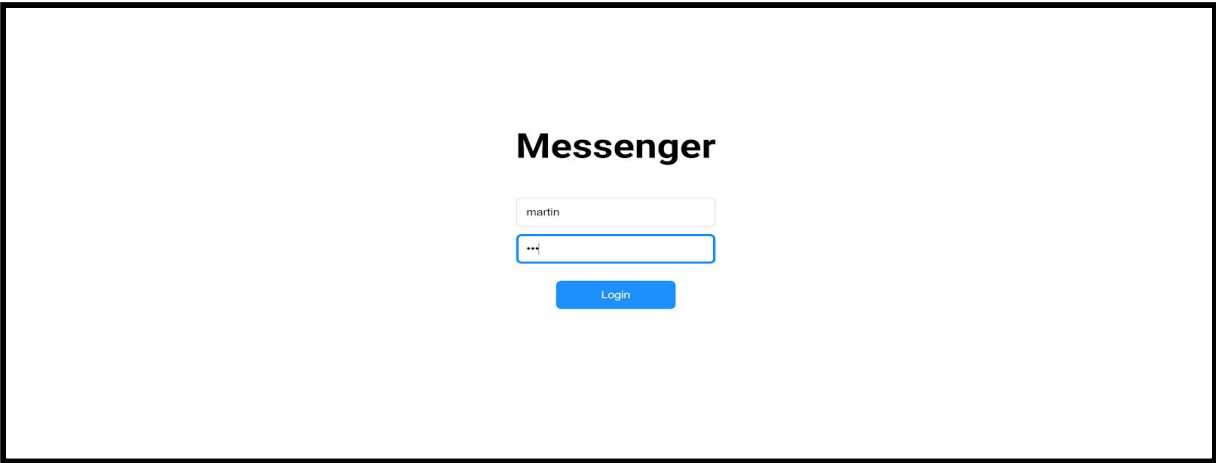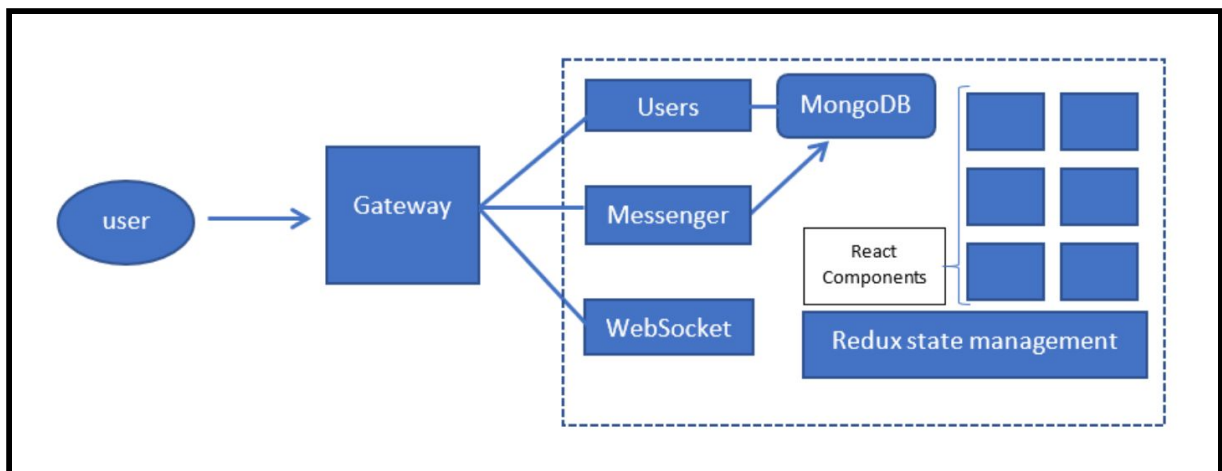
User Search Page

Chat Room

When testing the chat with two different user accounts PLEASE use incognito or a different browser to log in.

Architecture

VI. Self- Reflection on the Development process during the final project

The hardest part of this project was creating an efficient communication between all the team members. Given the fact that we were initially two groups that teamed up together, all of us did not know what to expect for half of the other team members. On top of that, because we did not have a team leader, and we did not know the strength and coding level of the other members, it was hard to assign a specific task to someone. Although the majority of us were sort of cooperative in terms of agreeing for what was best for our project, in some, we were lost at how to proceed with the outline of the project. As for the coding, designing the conditional rendering for all the components and states was sort of complicated especially for the class messenger.js that's located within the pages folder. Another difficult class was figuring out how to set up the database for messenger.js since we were trying to connect users together given the id of the chat.

The last challenge for this project was using docker. One of our team members spent days and hours trying to understand and figure out how to create an image of all the files that we needed to run in order to deploy our project in AWS. After too many trials and errors, docker worked.

VII. Project Conclusion

Given the time constraints that we had, we managed to fix other issues that we had during the demo. For instance, we used to have a chat_ids and not the name of the user that you were chatting with on the sidebar. Moreover, each time that we

created a chat with a new user, we needed to refresh the page in order to make the changes could appear, but there's no need to do that anymore. If we had the chance to get more time to work on this project, we will definitely try to add another option where users can chat with more than one person. Overall, we all feel like we did a very good and decent job with this project. We are happy with the outcome.