**JIVE CHAT – React JS (Redux, Styled Components & Firebase Hooks)**

npx create-react-app jive-chat –-template redux

*Setup* Firebase: create Firebase.js and copy the config codes rom firebase

Get the counterlslcie.js and move it out to features folder then delete the counter folder

Search react router and install it: npm install react-router-dom

Install and import material ui core and material ui icons

Installing styled components

Go to website > click the documentation > installation

Go to package.json and grab the styled components codes

Install the vscode styled components from the vscode itself

Create the App Header with styled components

Create the App Body with styled components

Create Sidebar.js and SidebarOption.js with styled components

Create the **addChannel and selectChannel** functions and let it be the functions for clicking the SidebarOpionContainer on the SidebarOption.js

**Go to firebase,.js and install firebase: npm i firebase** (connecting front-end to firebase database)

Go to firebase > firestore (database – NoSQL Structure) > Start in test mode

\*\*Create Google authenticator below the firebase database and authentication

const provider = new firebase.auth.GoogleAuthProvider();

Go to firebase > Authentication > Get Started > Google > Enable

**Adding the channel:** (Note: following steps only applies to demo version of the app)

*Firebase > Firestore > rules > remove the if statement for allowing users to read and write*

*Go back to Firebase > Firestore > Data*

*Add codes :*

const addChannel = () => {

    const channelName = prompt("Please enter the Channel Name");

    if (channelName) {

      db.collection("rooms").add({

        name: channelName,

      });

    }

  };

  const selectChannel = () => {};

***Install Firebase hooks*** *>* npm install --save react-firebase-hooks

Pull up data from database (firestore)

Go back to sidebar.js

Pull data

const [channels, loading, error] = useCollection(db.collection("rooms"));

Go back to sidebar option and configure the add the id as props

“Selecting channel using Redux” (*pushing he channel ID to redux*)”

>Rename counterSlice.js to appSlice.js > Go to app.js and import appReducer from "../features/appSlice";

>On the appSlice.js, rename all counterSlice, counter to appSlice, app codes then inititil state to roomId

export const appSlice = createSlice({

  name: "app",

  roomId: null,

Get rid of the “Increment Async codes”, “decrement and incrementByamount functions”, Let the “increment function” stays.

Import other necessary codes.

Go back to /sidebaroption.js and input the select channel declarations and functions.

**“THE CHATBOX SECTION”**

* Create a **chat.js** component > input codes and styled components
* Create **chatmessages.js and chatinput.js** component then call the data from firebase and return it to the app
* Create **Message.js** component and put it inside the **ChatMessages** component
* Once done, Setup the **Auto scroll** functionality for new messages coming in by adding **<ChatBottom/>** before the end of the **Chatmessages** component.
  + Add **chatRef** for the newly created **ChatBottom** and create a **useEffect** to handle scrolling up for new messages
  + Pass the **Chatref** to inside the **chatinput** component

**“USER AUTHENTICATION”**

1. Go to app.js > import and declare **useAuth** hooks and **auth** fromfirebase
2. Add the **Login** component and the if-else statement inside the router.
3. Create the **Login.js** > input codes and styles
   1. Input google authentication inside the Login function using these codes:
4. const signIn = (e) => {
5. e.preventDefault();
6. auth.signInWithPopup(provider).catch((error) => alert(error.message));
7. };

**“Getting the information of the current logged in user”**

* Go to Header.js > Inside function, input: const [user] = useAuthState(auth);
* Inside HeaderAvatar, declare the photo and the name of the user.
* Implement the same things mentioned above for the other necessary components e.g chatInput.js

“Extra: Loading Screen”

Install react-spinkit