Documentatie

1. Prezentare generala

Aplicatia presupune interactiunea utilizatorilor intre ei, prin intermediul unui chat unde pot comunica pe baza unui cont.

Aplicatia se structureaza in doua pagini: una in care utilizatorul se poate inregistra sau se poate autentifica si o pagina in care utilizatorul poate comunica cu alti utilizatori.

2. Structura aplicatiei

Fisierul de baza este App.jsx unde se aduc in pagina toate componentele chat-ului si totodata se fac urmatoarele verificari: daca datele sunt in curs de incarcare in pagina se va afisa mesajul “Loading…” pe ecran, iar daca utilizatorul este conectat se va afisa pagina chat-ului, altfel se va afisa pagina de inregistrare/autentificare.

In acest fisier sunt incluse 5 componente care se regasesc in folderul components: chat, detail, list, login si notification.

Componenta chat este componenta in care sunt afisate mesajele si este impartita in 3 sub-componente: parte de sus unde se afla numele utilizatorului si poza acestuia cu care se interactioneaza, impreuna cu 3 butoane de call, video call si info; in partea din mijloc sunt mesajele care se transmit intre cei doi utilizatori; iar in partea de jos este bara unde se scriu mesajele impreuna cu 5 butoane, send cu care trimitem mesajul, emoji cu care adaugam emoji, microfon in cazul in care transmitem mesaj vocal, camera cu care sa facem o poza, si image cu care se trimit poze.

Componenta detail este componenta unde se pot face anumite setari in legatura cu chat-ul aflat deschis, se pot vedea pozele in care au fost trimise pana acum in acest chat si se regasesc si doua butoane de block user si log out.

Componenta list este componenta unde se pot vizualiza toti utilizatori cu care a interactionat utilizatorul impreuna cu numele si poza acestora si se mai poate observa o parte de informatii despre utilizator impreuna cu optiuni. Aceatsa componenta este impartita in doua subcomponente: chatList si userInfo. Componenta chatList este componenta care afiseaza lista de utilizatori cu care s-a interactionat si mai contine o subcomponenta addUser care prin intermediul unui buton plus, deschide un modal in care se poate cauta un nou utilizator si se poate incepe interactiunea cu acesta. Componenta userInfo este componenta care afiseaza informatii despre utilizatorul autenficat, poza si numele acestui, impreuna un mic meniu compuse din 3 butone, 3 puncte, camera si un creion.

Componenta login, deschide pagina de start a aplicatiei, in care utilizatorul isi poate creea un cont prin completarea unui form cu 3 inputuri username, email si password si adaugarea unei poze de profil si totodata se poate autentifica prin completarea email-ului si a parolei.

Componenta notification afiseaza mesajele de eroare in cazul in care apar erori la completarea inputurilor din pagina de autentificare/inregistrare.

In folderul lib se regasesc 4 fisiere de configuratii. Fisierul firebase este folosit pentru initializarea conexiunii la o baza de date firebase, impreuna cu initializare unor module esentiale in lucrul cu bazele de date firebase, ca de exemplu firestore care este o baza de date NoSQL, dar a carei date sunt stocate in colectii. In fisierul userStore se creeaza un “store” de utilizator folosind biblioteca “zustand” pentru gestionarea starii si utilizeaza firestore pentru a prelua informatiile unui utilizator dintr-o baza de date. In fisierul chatStore se creeaza un “store” de chat folosind biblioteca “zustand” pentru gestionarea starii, de asemeneaza se utilizeaza firestore pentru a prelua informatiile despre acel chat din baza de date, se permite schimbarea chat-ului si se verifica daca un utilizator este blocat. In fisierul upload se creeaza o functie asincrona care incarca un fisier in firebase si returneaza un URL pentru acesta.

3. Functionalitatea aplicatiei

In momentul accesarii aplicatiei de catre un utilizator nou, se va incarca in pagina clasa loading din fisierul App, care este afisata datorita verificarii daca datele au fost sau nu incarcate. Dupa aceasta verificare se face verificarea daca utilizatorul este autentificat sau nu, iar in cazul unui utilizator nou va fi redirectionat in pagina de inregistrare/autentificare.

In pagina de autentificare/inregistrare utilizatorul isi creeaza un cont, prin completarea a 3 inputuri si incarcarea unei imagini pentru utilizator. Aceste inputuri sunt incluse intr-un form in fisierul login care apeleaza o functie asincrona handleRegister in care prin intermediul functiei FormData preuia datele introduse in formular, iar prin intermediul functiei createUserWithEmailAndPassword din firebase se creeaza un nou utilizator cu datele introduse in formular, apoi prin intermediul functiei upload, poza incarcata de utilizator in formular va fi transformata in URL, iar functie setDoc are rolul de a crea un nou document in colectia “users” din firestore, stocand informatiile utilizatorului si creeaza si un nou document chats din colectia “userchats” din firestore unde vor fi salvate mesajele transmise de acest utilizator.

Dupa ce utilizatorul si-a creat contul cu formularul de register, trebuie sa se conecteze prin intermediul formularului de login unde v-a completa campurile de email si parola. Aceste inputuri sunt continute intr-un form in fisierul login care, la submit, apeleaza o functie asincrona handleLogin care prin intermediul functiei FormData preia datele introduse in inputuri, iar prin intermediul functiei Object.fromEntries extrage campruile email si password din obiectul FormData, dupa care se incearca autentificarea utilizatorului prin intermediul functiei signInWithEmailAndPassword folosind adresa de email si parola. In cazul in care nu se reuseste se va afisa un mesaj de eroare, altfel se va face redirect la pagina cu chat-ului.

Dupa autentificare utilizatorul va fi redirectionat in pagina chat-ului din fisierul App insa se mai face urmatoarea verificare, daca utilizatorul nu are deschisa nici o conversatie cu un alt utilizator atunci acesta nu va putea vedea componentele de chat si detail ci doar componenta list. Pentru a vedea si celelalte componente utilizatorul va trebui sa inceapa interactiunea cu un alt utilizator, pentru aceasta trebuie sa actioneze butonul plus. Acest buton plus are atasta o variabila addMode care reprezinta starea modalului, care este setata initial pe false adica modalul este inchis, iar in momentul in care se apasa acel buton se apeleaza la onclick functia setAddMode cu o valoare opusa valorii lui addMode, adica modalul se deschide. Daca addMode este false se va afisa plus, iar daca addMode este true se va afisa minus. Odata deshis, modalul contine doar o bara de cautare si un buton search, utilizatorul va completa acel input cu username-ul utilizatorului cu care doreste sa interactioneze, dupa care va apasa butonul search. Odata actionat butonul search acesta va apela o functie asincrona handleSearch care in primul rand va creea un nou obiect cu valorile introduse in input prin intermediul functie FormData, apoi prin intermediul functiei query va creea o interogare intre valoarea introdusa in inputu si o referinta catre colectia “users” din baza de date firestore, adica cauta utilizatorul cu username-ul specificat. Daca acest utilizator este gasit atunci va aparea o clasa care contine numele si poza utilizatorului respectiv, impreuna cu un buton “Add User”. La actionarea butonului “Add User” se va apela o functie asincrona handleAdd care prin intermeidul functiei doc si setDoc creeaza si adauga un nou document in colectia “chats” din firestore, apoi actualizeaza documentul din colectia “userchats” prin intermediul functiei updateDoc pentru utilizatorul curent si adauga o noua conversatie in lista “chats” a utilizatorului. Apoi actualizeaza documentul din colectia “userchats” a utilizatorului ales si adauga o noua conversatie in lista “chats” a utilizatorului ales.

Dupa adaugarea unui alt utilizator, va putea incepe o conversatie cu acesta si va putea vizualiza si celelalte componente chat si detail.

Pentru transmiterea de mesaje utilizatorul va completa inputul cu mesajul pe care doreste sa-l transmita si va actiona butonul send. Odata apasat butonul send, va apela o functie handleSend care mai intai verifica daca mesajul este gol sau nu, daca este gol se opreste si nu face nimic, daca nu este gol se verifica daca utilizatorul a selectat o imagine pe care o incarca in firebase storage folosind functia upload. Apoi se actualizeaza documentul corespunzator chat-ului curent din colectia “chats” din firebase prin intermediul functiei updateDoc, se adauga un nou mesaj in lista de mesaje a chat-ului folosind functia arrayUnion si actualizeaza si documentul “userchats” pentru utilizatorul caruia s-a transmis mesajul, se actualizeaza informatiile despre conversatia curenta: ultimul mesaj trimis, starea de vizualizare a mesajului si data ultimei actualizari a conversatiei. Dupa trimiterea mesajului, starea pentru textul mesajul si imaginea sunt resetate pentru a pregati interfata pentru un nou mesaj.

Pentru a schimbat chat-ul cu un alt utilizator, utilizatoru v-a actiona chat-ul cu utilizatorul cu care doreste sa interactioneze si odata apasat pe acel chat se va apela functia handleSelect. Funcția începe prin a verifica dacă există un utilizator autentificat (currentUser). Dacă utilizatorul nu este autentificat (currentUser este null sau undefined), funcția se încheie imediat prin instrucțiunea return. Se creează o referință către documentul userchats al utilizatorului curent în Firestore folosind doc(db, "userchats", currentUser.id). Apoi, se obține o imagine statică a acestui document utilizând getDoc(userChatsRef), iar datele documentului sunt extrase utilizând userChatsSnapshot.data(). Funcția creează o nouă listă updatedChats bazată pe datele existente despre conversații ale utilizatorului. Fiecare element din listă este verificat în raport cu chat.chatId. Dacă chat.chatId coincide cu identificatorul unui chat din listă, acest chat este marcat ca văzut prin setarea isSeen: true. Altfel, elementul este returnat nemodificat. Utilizând updateDoc(userChatsRef, { chats: updatedChats }), lista actualizată de conversații este încărcată înapoi în documentul userchats al utilizatorului curent din Firestore. Astfel, starea de vizualizare a mesajelor este actualizată în baza de date. Prin apelarea funcției changeChat(chat.chatId, chat.user), aplicația actualizează chat-ul curent selectat cu cel specificat în parametrii funcției (chat.chatId și chat.user). Dacă apare o eroare în timpul actualizării documentului Firestore, aceasta este înregistrată în consola browserului utilizând console.log(err).

In cazul in care utilizatorul doreste sa blockeze un alt utilizator acest poate actiona butonul “Block user” care apeleaza functia handleBlock. Funcția începe prin a verifica existența utilizatorului (user). Dacă utilizatorul nu există (este null sau undefined), funcția se încheie imediat prin instrucțiunea return. Se creează o referință către documentul utilizatorului curent din Firestore folosind doc(db, "users", currentUser.id). Acest lucru presupune că currentUser.id este identificatorul unic al utilizatorului curent în baza de date Firestore. În funcție de starea utilizatorului vizualizat (user) și de starea de blocare a receptorului (isReceiverBlocked), funcția decide dacă utilizatorul trebuie adăugat sau eliminat din lista de utilizatori blocați a utilizatorului curent. Asta se realizează utilizând funcții Firestore arrayUnion și arrayRemove. Utilizând updateDoc(userDocRef, { blocked: ... }), lista actualizată de blocări este încărcată înapoi în documentul utilizatorului curent din Firestore. Apelând funcția changeBlock(), aplicația poate reacționa la schimbarea în starea de blocare a utilizatorului și poate actualiza interfața în consecință. Dacă apare o eroare în timpul actualizării documentului Firestore, aceasta este înregistrată în consola browserului utilizând console.log(err).

In momentul in care utilizatorul doreste sa iasa din cont atunci va actiona butonul “Sin Out” care va apela functia signOut din libraria auth.