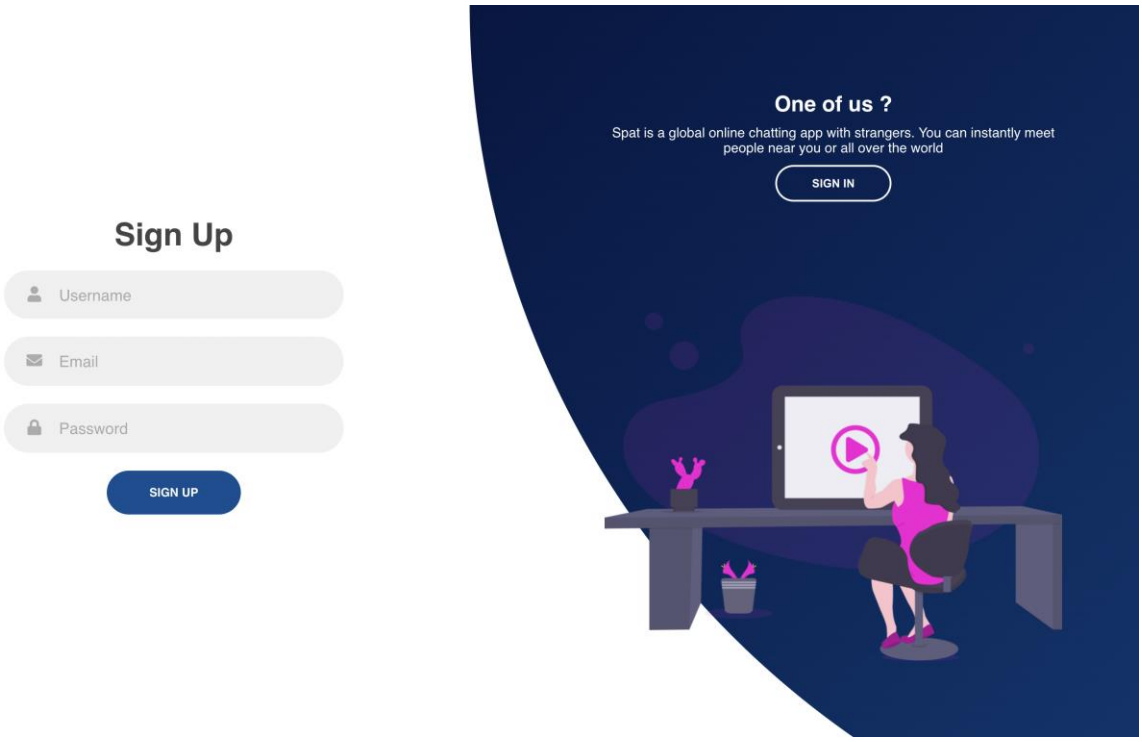
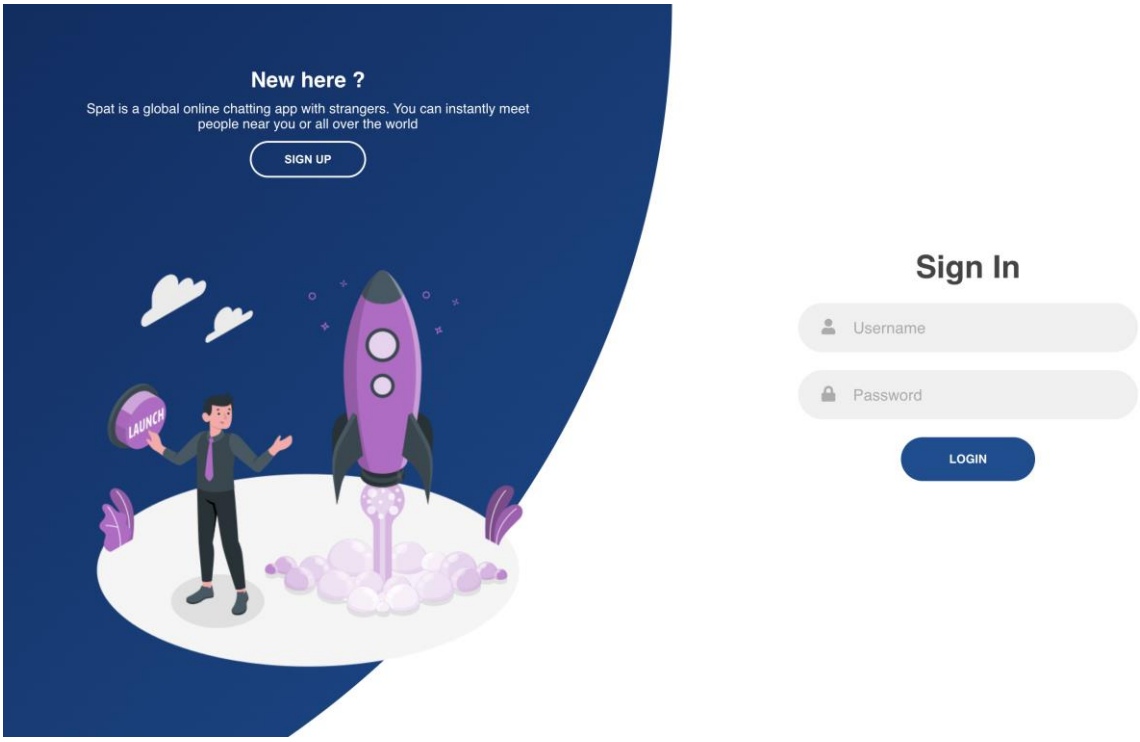


Sign up page



Enter email id and password and username to login
Sign in page

Enter email id and password login

For sign up and sign in we are using firebase
Firebase connection is done in two file
1 config.js
this is basically for connection info api key
2 firebaseconnector.js
this file is used to define what all services we are using of firebase and config them in this file

Onsubmitsignup function is used for storing signup info into firebase

Onsubmitsignin function is used for storing signin info into

firebase
This page is used to get info of all the users trying to connect to users on website

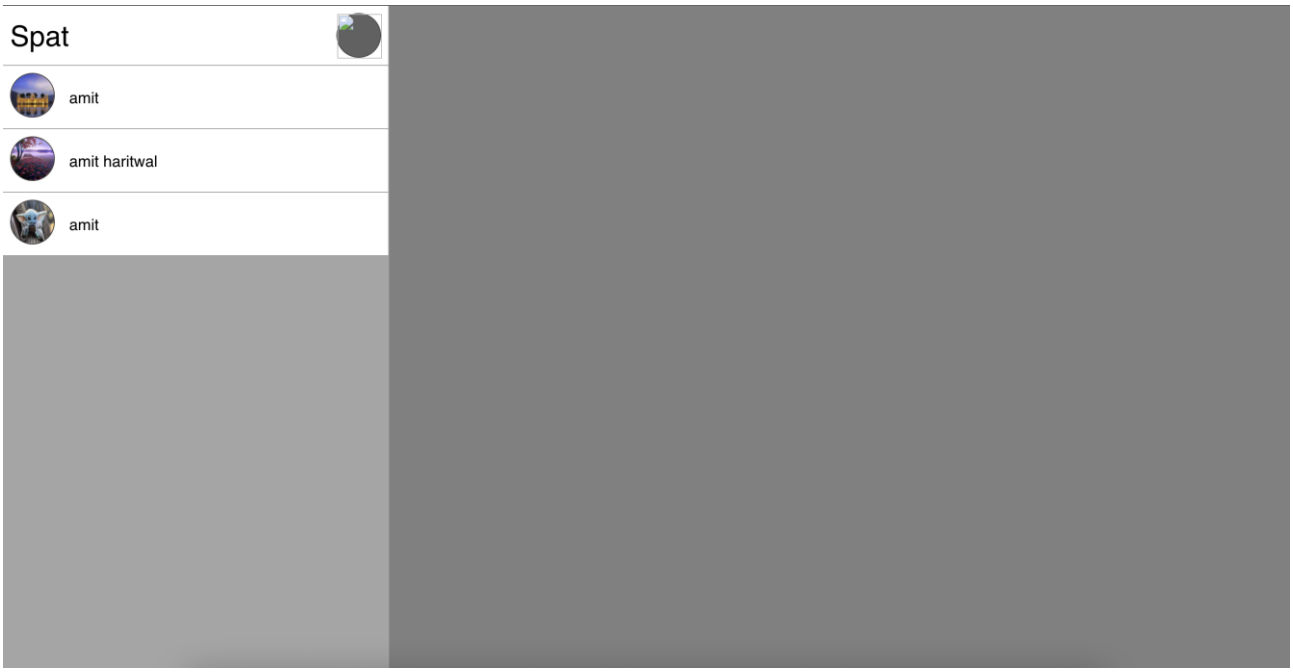
Addpartner.js

this file is used to find all the users which are not connected

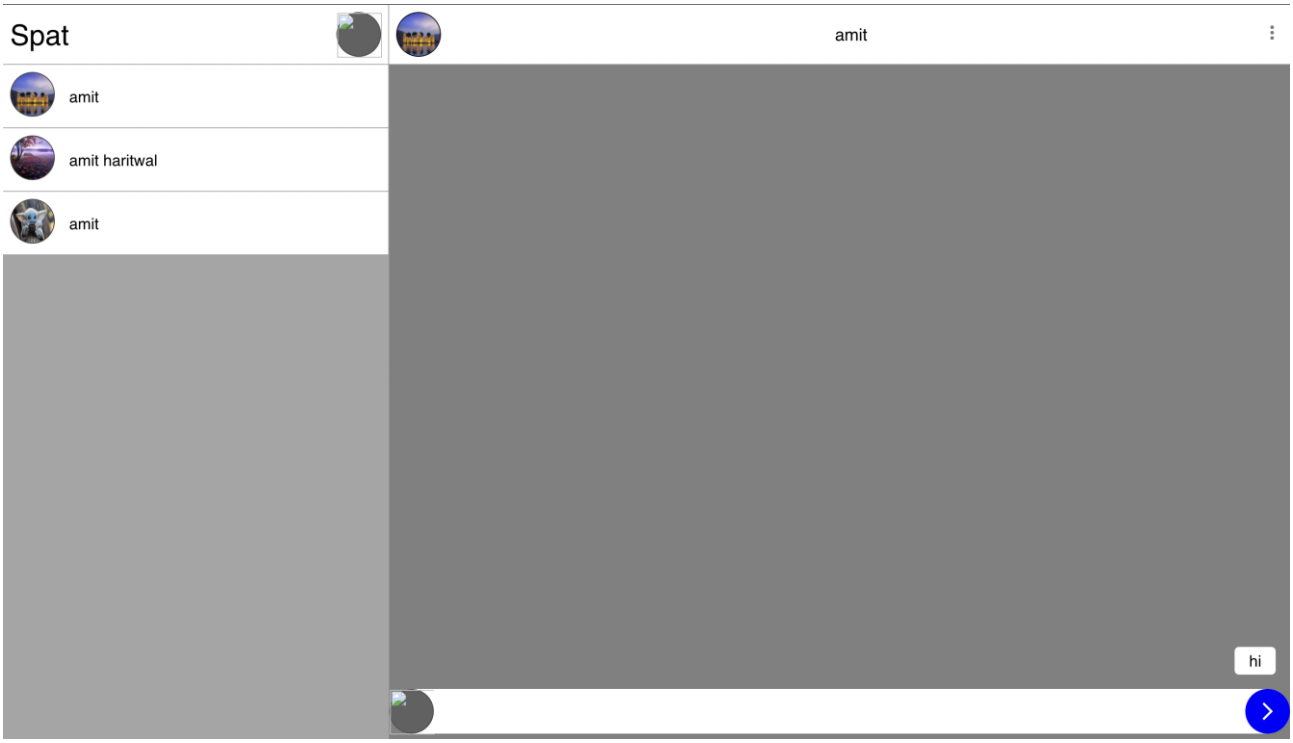
```
const [userData, setUserData] = useState([]);
const userInfo = useAppSelector(selectUserDetails);
const addSubscriber = async (id) => {…
};
const getUserList = async () => {
  var info = userInfo.subscribed;
  var sub = [];
  if (!info) info = [];
  for (var i = 0; i < info.length; i++) {
    sub.push(info[i].suid);
  }
  sub.push(userInfo.id);
  const q = query(collection(db, "users"));
  var temp2 = [];
  const querySnapshot = await getDocs(q);
  querySnapshot.forEach((doc) => {
    var temp = doc.data();
    if (!sub.includes(temp.userId)) {
      temp2.push(temp);
    }
  });

  setUserData(temp2);
};
useEffect(() => {
  if (!userData.length) getUserList();
});
return (
  <>
```


to you

There are two function in this file one is getuserlist which is used to get all the info of users which are not connected to you on this platform and second is addSubscriber which is trigged when you click on hand shake button which is used to connect both of the users


Spat
amit
amit haritwal
amit

This page is the main page on the left side this list of connected users are there to which you are connected and to chat to them



you have to click on that user only

This is the right section which appears when you click on particular user on top you get the users profile with there name and on the bottom you have message input box and there is profile on left and submit button on right when you write and send

a message then it will appear on right side and incoming message will appear on left side.

```
5  import AddPartner from "./AddPartner";
6
7  import { db } from "./firebaseconnector";
8  import { useAppDispatch, useAppSelector } from "./Hooks";
9
10 import { CloseIcon, MenuIcon, SendMessageIcon } from "./icons";
11 import { addChatDetails, selectUserDetails } from "./userreducer";
12
13 var mqtt = require("mqtt");
14 var client = mqtt.connect("ws://13.40.23.172:8888");
15
16 if (client.connected) {
17   console.log("connected");
18 } else {
19   console.log("sorry not able to connect");
20 }
21 function Chat() {
22   const input = useRef(null);
23   const userInfo = useAppSelector(selectUserDetails);
24   const [partnerPopup, setPartnerPopup] = useState(true);
25   const dispatch = useAppDispatch();
26   const [message, setMessage] = useState("");
27   const [messages, setMessages] = useState([]);
28   const [allUsers, setAllUsers] = useState([]);
29   const [isSubscribed, setIsSubscribed] = useState(false);
30   const [currentState, setCurrentState] = useState({
31     name: "",
32     profileUrl: "",
33     currentChannelName: "",
```

Chat.jsx is the file in which all the this are done first we connect to Matt broker using mqtt module and the link on line no 14 is the place where we connect to them

getconnectioninfo is used to get all the info of your connection with the help of user details which is stored in redux in details we

```
const getConnectionInfo = async () => {
  if (userInfo.subscribed) {
    var info = userInfo.subscribed;
    var tosub = [];

    for (var i = 0; i < info.length; i++) {
      tosub.push(info[i].suid);
    }
    const q = await query(
      collection(db, "users"),
      where("userId", "in", tosub)
    );

    const querySnapshot = await getDocs(q);
    var sideusers = [];
    querySnapshot?.forEach(async (docdetails) => {
      var data = docdetails.data();
      for (var i = 0; i < info.length; i++) {
        if (info[i].suid === data.userId) {
          data.scid = info[i].scid;
          break;
        }
      }
      sideusers.push(data);
    });

    setAllUsers(sideusers);
  }
};
```

are getting userId which is used to get user full details and connection id which is unique key which is combination of your id

and your connection id by which we can identify who to show

```
};

useEffect(() => {…
}, [userInfo, dispatch]);

useEffect(() => {
  if (!isSubscribed && userInfo.subscribed) {
    setIsSubscribed(true);
    var info = userInfo.subscribed;
    var tosub = [];
    if (info) {
      for (var i = 0; i < info.length; i++) {
        tosub.push(info[i].scid);
      }
    }
    client.subscribe(tosub);
    console.log(client);
  }
}, [setIsSubscribed, isSubscribed, userInfo]);

useEffect(() => {…
```

chat

This useeffect is used to subscribe to all the connection whose details we got from get connection info

```
143
144  useEffect(() => {
145    if (userInfo.chat[currentState.currentChannelName]) {
146      var temp = [];
147      for (
148        var i = 0;
149        i < userInfo.chat[currentState.currentChannelName].length;
150        i++
151      ) {
152        temp.push({
153          message: userInfo.chat[currentState.currentChannelName][i].message,
154          isSelf: userInfo.chat[currentState.currentChannelName][i].isSelf,
155        });
156      }
157      setMessages(temp);
158    } else setMessages([]);
159  }, [currentState.currentChannelName, userInfo.chat]);
160
161  const [isMenuClosed, setIsMenuClosed] = useState(true);
162  return (
163    <div className="main">
```

This use effect is used to push your chat to redux where we store chat. So that there is smooth transition between users

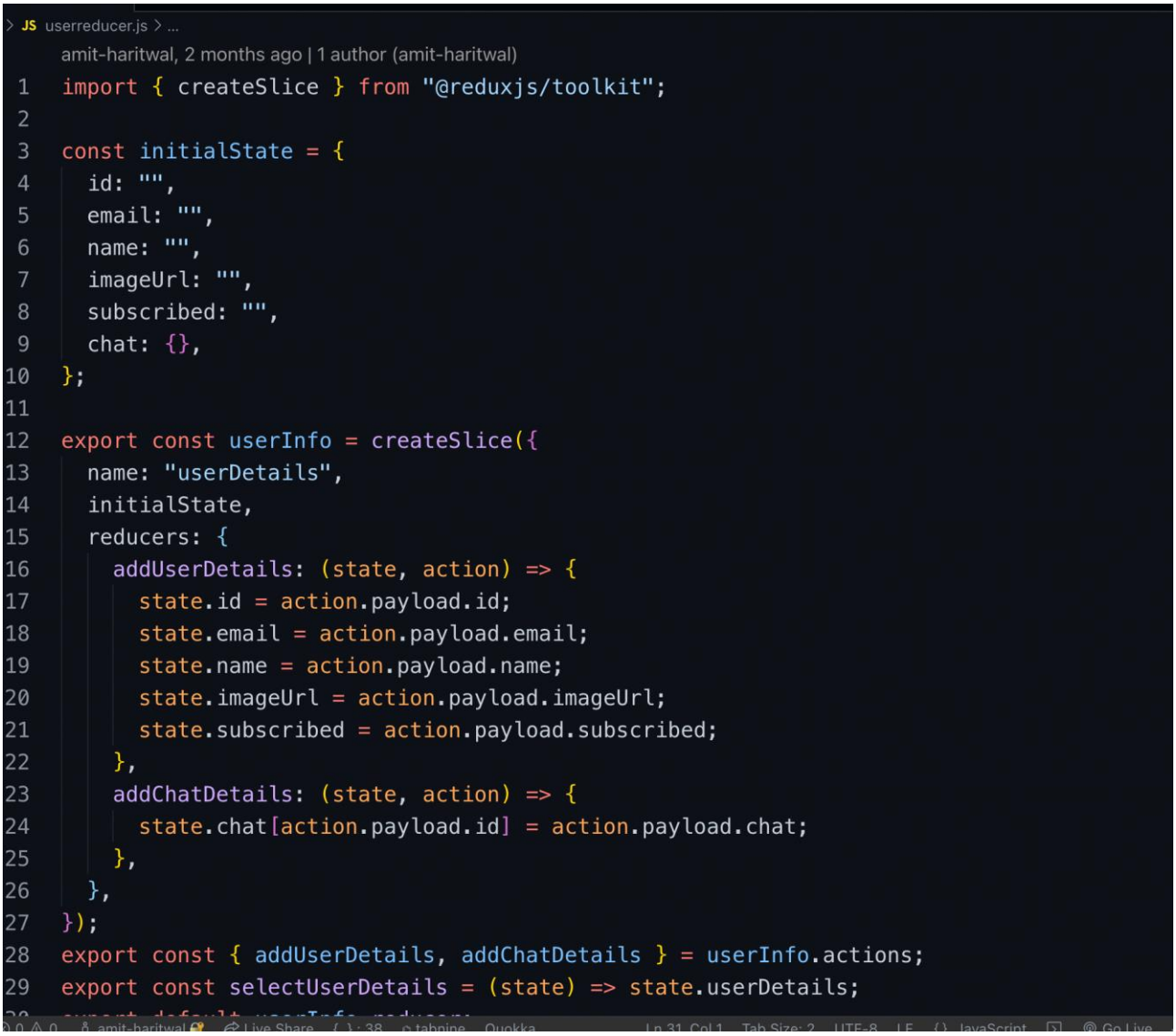
```
32      profileUrl: "",
33      currentChannelName: "",
34    });
35
36    const sendMessage = async () => {
37      console.log(currentState.currentChannelName);
38      await client.publish(
39        currentState.currentChannelName,
40        message + userInfo.id
41      );
42      console.log("senging message: " + message, currentState.currentChannelName);
43      setMessage("");
44    };
45
46 ›  const getConnectionInfo = async () => {…
74    };
75
76 ›  useEffect(() => {…
114   }, [userInfo, dispatch]);
115
116 ›  useEffect(() => {…
```

Send message is used to send message to that subscriber on which you recently clicked on left side

```
src > Chat.jsx > ...
76    useEffect(() => {
77      client.on("message", (topic, message) => {
78        var note;
79        note = message.toString();
80
81        var temp = note.substring(0, note.length - 36);
82
83        setMessages((prevState) => {
84          var data = prevState;
85          if (note.substring(note.length - 36) === userInfo.id)
86            data.push({ message: temp, isSelf: true });
87          else data.push({ message: temp, isSelf: false });
88          return data;
89        });
90        var data = [];
91        if (userInfo.chat[topic]) {
92          for (var i = 0; i < userInfo.chat[topic].length; i++) {
93            data.push({
94              message: userInfo.chat[topic][i].message,
95              isSelf: userInfo.chat[topic][i].isSelf,
96            });
97          }
98        }
99        if (note.substring(note.length - 36) === userInfo.id) {
100         data.push({ message: temp, isSelf: true });
101       } else {
102         data.push({ message: temp, isSelf: false });
103       }
104
105       dispatch(
106         addChatDetails({
107           id: topic,
108           chat: data,
109         })
110       );
111       setMessage(" ");
112       setMessage("");
```

This is the most important part for chat this is basically to get chat info for Matt broker and we identify that it is my message or others message so that according to that we can put that to left or right and push that to redux

Router.js is basic routing done in react



This is redux info so there are two reducer on of them is to store user info and another is to store chat info