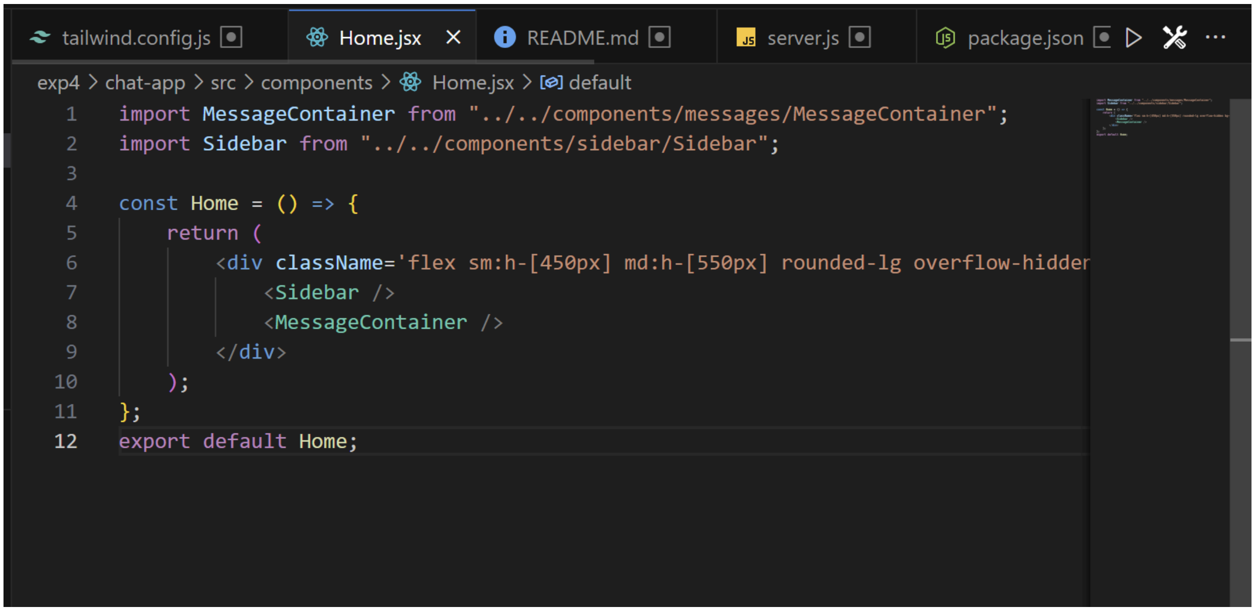
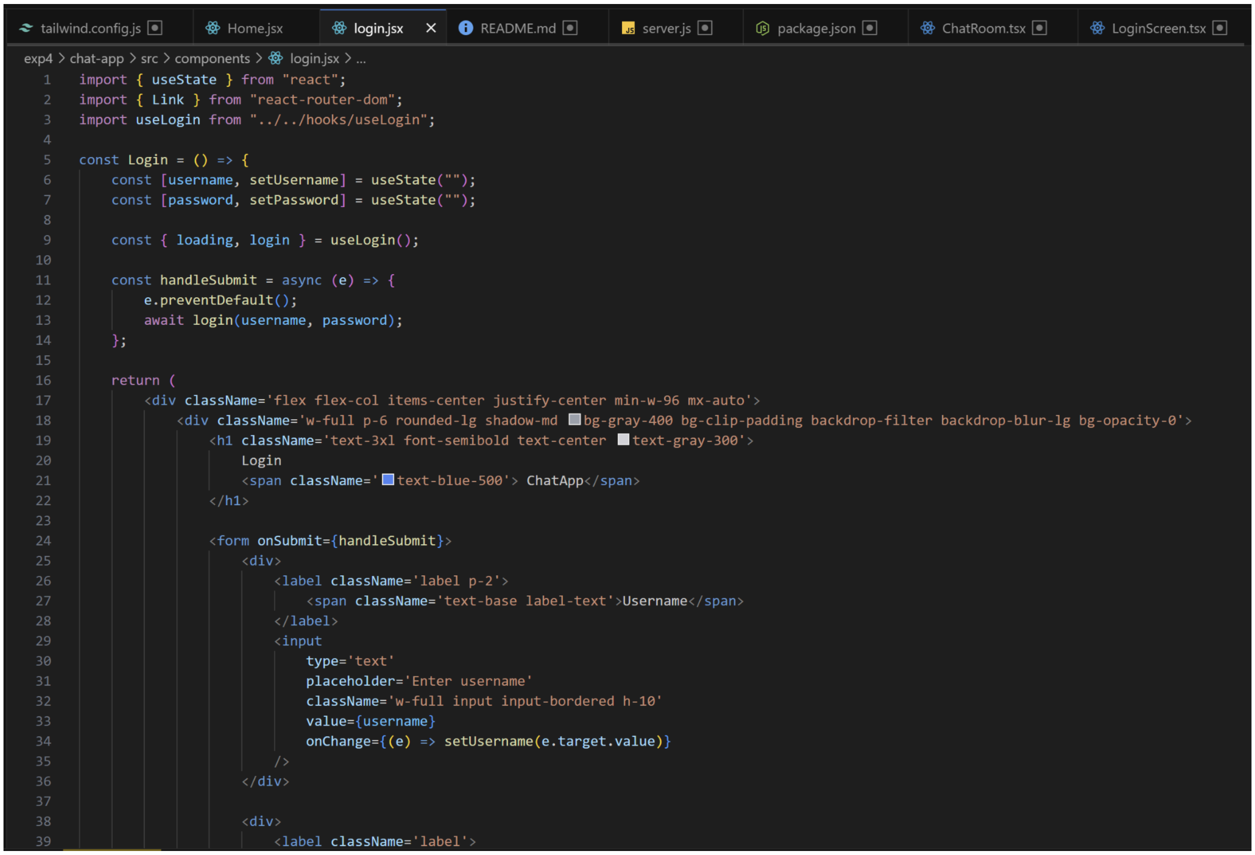
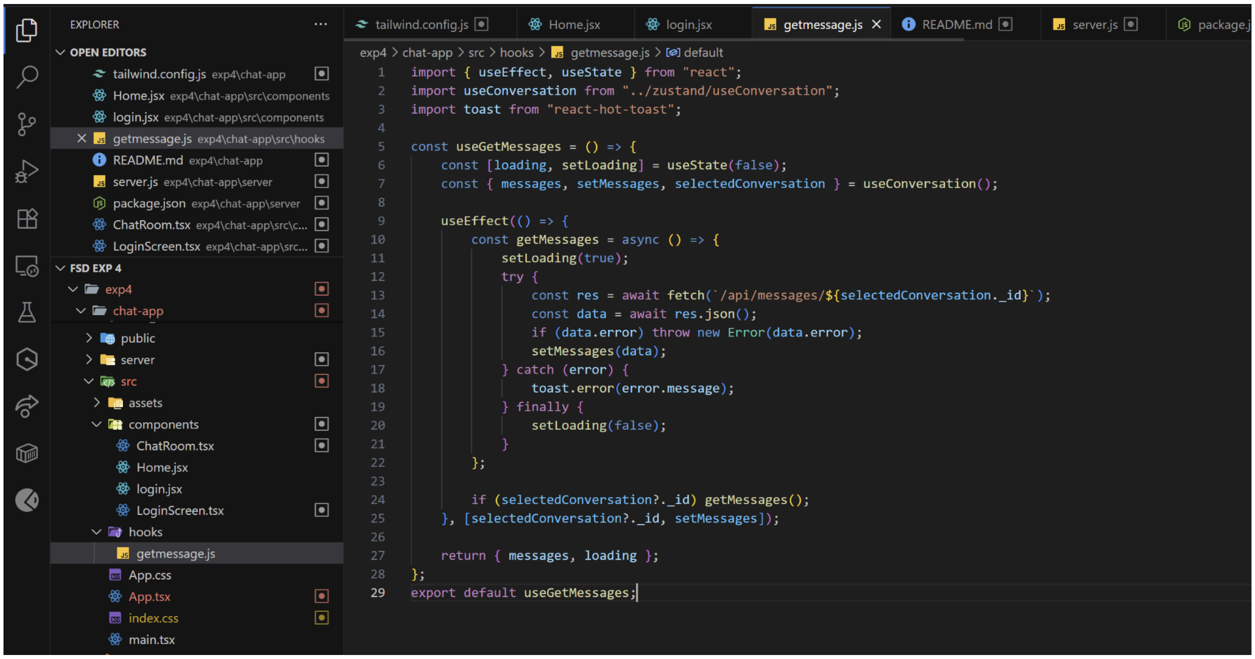


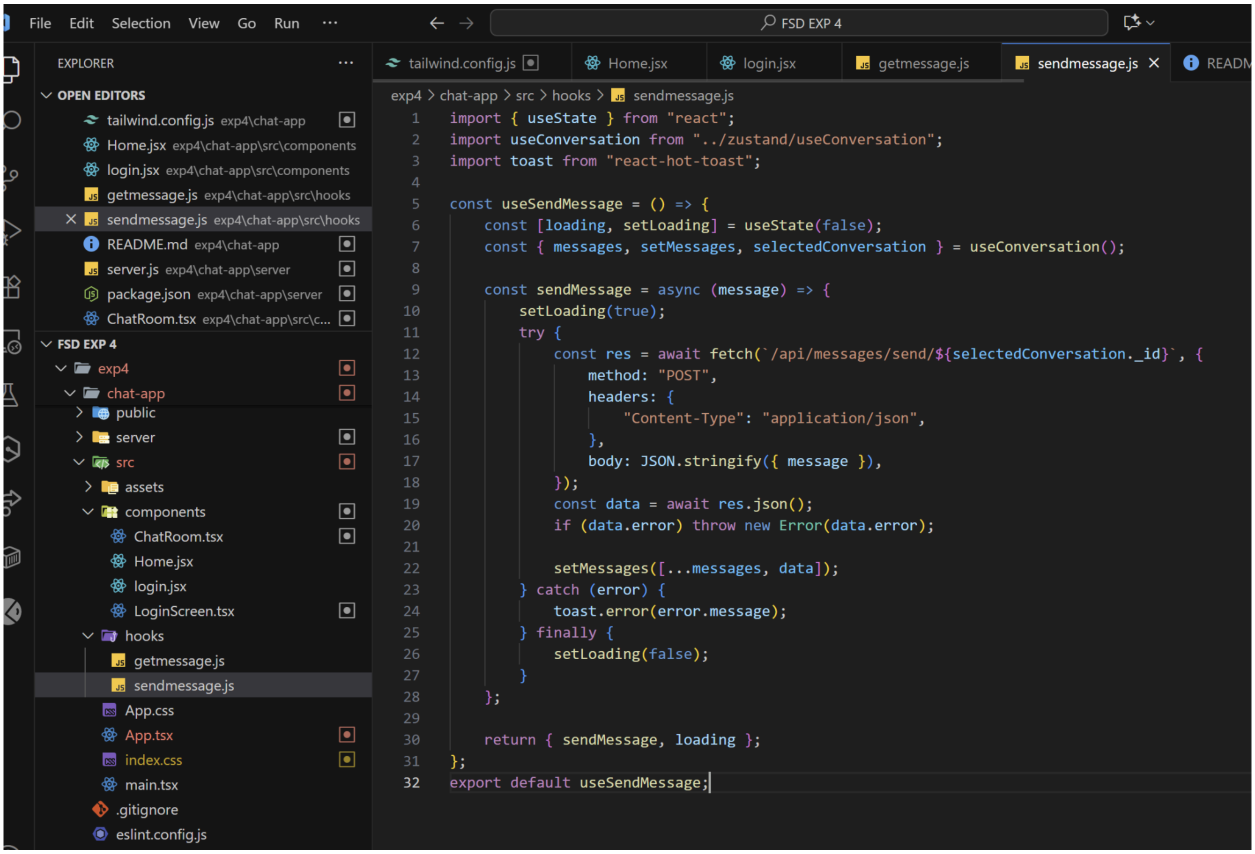
```jsx
import MessageContainer from "../../components/messages/MessageContainer";
import Sidebar from "../../components/sidebar/Sidebar";

const Home = () => {
    return (
        <div className='flex sm:h-[450px] md:h-[550px] rounded-lg overflow-hidder
            <Sidebar />
            <MessageContainer />
        </div>
    );
};
export default Home;
```



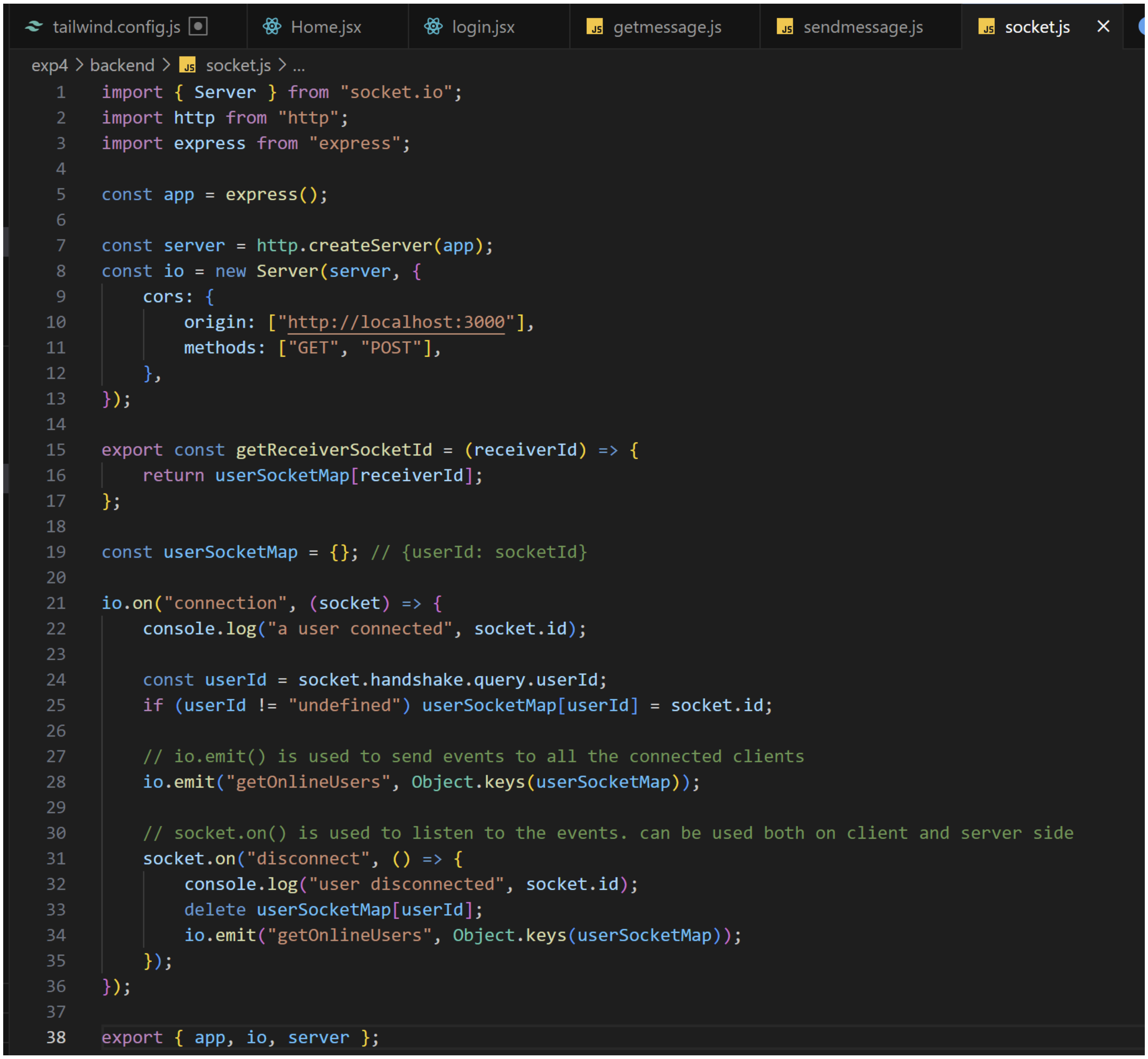
```jsx
import { useState } from "react";
import { Link } from "react-router-dom";
import useLogin from "../../hooks/useLogin";

const Login = () => {
    const [username, setUsername] = useState("");
    const [password, setPassword] = useState("");

    const { loading, login } = useLogin();

    const handleSubmit = async (e) => {
        e.preventDefault();
        await login(username, password);
    };

    return (
        <div className='flex flex-col items-center justify-center min-w-96 mx-auto'>
            <div className='w-full p-6 rounded-lg shadow-md bg-gray-400 bg-clip-padding backdrop-filter backdrop-blur-lg bg-opacity-0'>
                <h1 className='text-3xl font-semibold text-center text-gray-300'>
                    Login
                    <span className='text-blue-500'> ChatApp</span>
                </h1>

                <form onSubmit={handleSubmit}>
                    <div>
                        <label className='label p-2'>
                            <span className='text-base label-text'>Username</span>
                        </label>
                        <input
                            type='text'
                            placeholder='Enter username'
                            className='w-full input input-bordered h-10'
                            value={username}
                            onChange={(e) => setUsername(e.target.value)}
                        />
                    </div>

                    <div>
                        <label className='label'>
```


EXPLORER
OPEN EDITORS
tailwind.config.js exp4\chat-app
Home.jsx exp4\chat-app\src\components
login.jsx exp4\chat-app\src\components
getmessage.js exp4\chat-app\src\hooks
README.md exp4\chat-app
server.js exp4\chat-app\server
package.json exp4\chat-app\server
ChatRoom.tsx exp4\chat-app\src\c...
LoginScreen.tsx exp4\chat-app\src...
FSD EXP 4
exp4
chat-app
public
server
src
assets
components
ChatRoom.tsx
Home.jsx
login.jsx
LoginScreen.tsx
hooks
getmessage.js
App.css
App.tsx
index.css
main.tsx
tailwind.config.js
Home.jsx
login.jsx
getmessage.js
README.md
server.js
package.j
exp4 > chat-app > src > hooks > getmessage.js > default

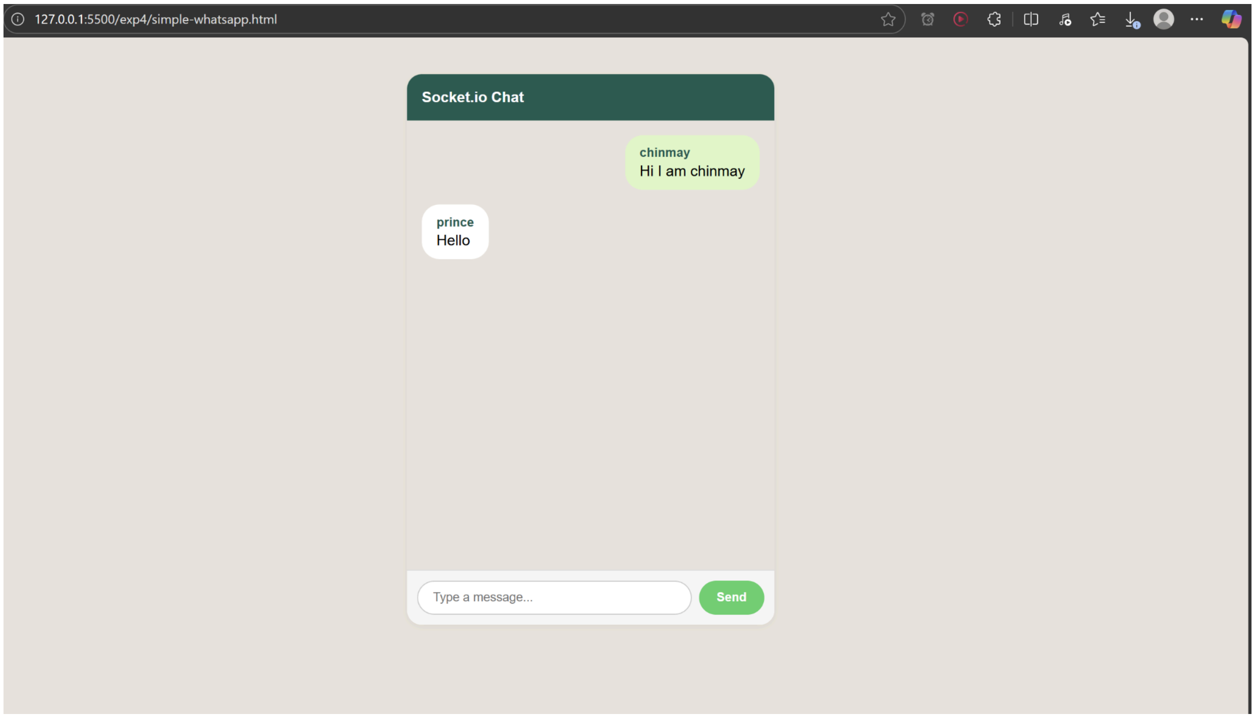
```javascript
import { useEffect, useState } from "react";
import useConversation from "../zustand/useConversation";
import toast from "react-hot-toast";

const useGetMessages = () => {
    const [loading, setLoading] = useState(false);
    const { messages, setMessages, selectedConversation } = useConversation();

    useEffect(() => {
        const getMessages = async () => {
            setLoading(true);
            try {
                const res = await fetch(`/api/messages/${selectedConversation._id}`);
                const data = await res.json();
                if (data.error) throw new Error(data.error);
                setMessages(data);
            } catch (error) {
                toast.error(error.message);
            } finally {
                setLoading(false);
            }
        };

        if (selectedConversation?._id) getMessages();
    }, [selectedConversation?._id, setMessages]);

    return { messages, loading };
};
export default useGetMessages;
```



```javascript
import { useState } from "react";
import useConversation from "../zustand/useConversation";
import toast from "react-hot-toast";

const useSendMessage = () => {
    const [loading, setLoading] = useState(false);
    const { messages, setMessages, selectedConversation } = useConversation();

    const sendMessage = async (message) => {
        setLoading(true);
        try {
            const res = await fetch(`/api/messages/send/${selectedConversation._id}`, {
                method: "POST",
                headers: {
                    "Content-Type": "application/json",
                },
                body: JSON.stringify({ message }),
            });
            const data = await res.json();
            if (data.error) throw new Error(data.error);

            setMessages([...messages, data]);
        } catch (error) {
            toast.error(error.message);
        } finally {
            setLoading(false);
        }
    };

    return { sendMessage, loading };
};
export default useSendMessage;
```



```javascript
import { Server } from "socket.io";
import http from "http";
import express from "express";

const app = express();

const server = http.createServer(app);
const io = new Server(server, {
    cors: {
        origin: ["http://localhost:3000"],
        methods: ["GET", "POST"],
    },
});

export const getReceiverSocketId = (receiverId) => {
    return userSocketMap[receiverId];
};

const userSocketMap = {}; // {userId: socketId}

io.on("connection", (socket) => {
    console.log("a user connected", socket.id);

    const userId = socket.handshake.query.userId;
    if (userId != "undefined") userSocketMap[userId] = socket.id;

    // io.emit() is used to send events to all the connected clients
    io.emit("getOnlineUsers", Object.keys(userSocketMap));

    // socket.on() is used to listen to the events. can be used both on client and server side
    socket.on("disconnect", () => {
        console.log("user disconnected", socket.id);
        delete userSocketMap[userId];
        io.emit("getOnlineUsers", Object.keys(userSocketMap));
    });
});

export { app, io, server };
```



Socket.io Chat

chinmay
Hi I am chinmay

prince
Hello