1. SETUP AND CLIENT :

Clone this repo : [safak/youtube at mern-social-app (github.com)](https://github.com/safak/youtube/tree/mern-social-app)

1. Npm install in both client and api
2. Previous code is for authentification and some fonctionallities we already saw before
3. Add .env file contain the url of mongodb inside api folder
4. Login mongo : ayoub ayoub90
5. Npm start on both sides
6. Create social account and enter your project (login : [ayoub.ammr@gmail.com](mailto:ayoub.ammr@gmail.com) ,ayoub1234)
7. U can Login with khalil99@gmail.com ,khalil99
8. Create messenger folder inside pages containing Messenger.jsx and messenger.css
9. In App.js add a route for messenger page :

<Route path="/messenger">

          {!user ? <Redirect to="/" /> : <Messenger />}

        </Route>

1. <http://localhost:3000/messenger>
2. Work on messenger UI :

* Add Top bar inside messenger component and those divs

 <>

      <Topbar />

      <div className="messenger">

        <div className="chatMenu">

          <div className="chatMenuwrapper">

          menu

          </div>

        </div>

        <div className="chatBox">

          <div className="chatBoxwrapper">

              box

          </div>

        </div>

        <div className="chatOnline">

          <div className="chatonlinewrapper">

              online

          </div>

        </div>

      </div>

    </>

Css :

.messenger {

    height: calc(100vh - 70px ) ;

    display: flex;

}

.chatMenu{

    flex : 3.5

}

.chatBox{

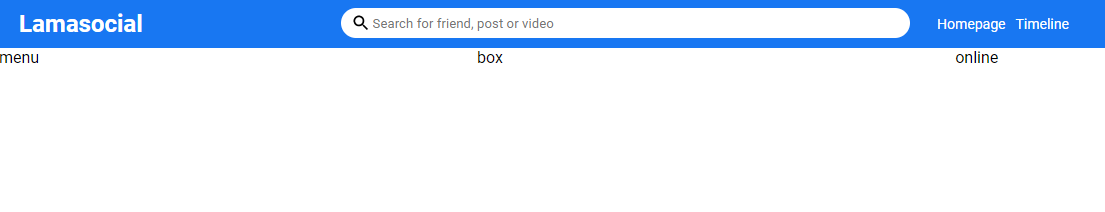
    flex : 3.5

}

.chatOnline{

    flex : 3

}



.chatMenuwrapper,.chatBoxwrapper , .chatOnlinewrapper {

padding : 10px ;

height: 100%;

background-color: aqua;

}

* Now we gonna work on chatMenu wrapper UI :

JSX :

<input type="text"

           placeholder="Search for friends"

            className="chatMenuInput" />

CSS :

*/\* CHAT MENU  WRAPPER SIDE \*/*

.chatMenuInput{

    width: 90%;

    padding : 10px 0 ;

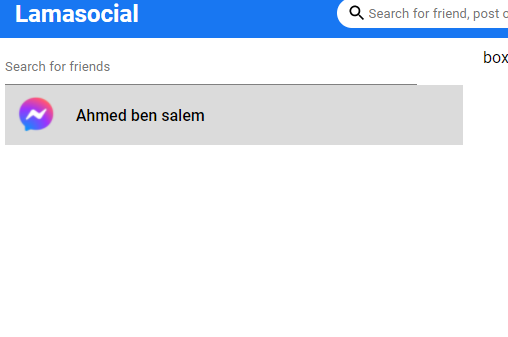
    border: none;

    border-bottom: 1px solid gray;

}

* Now add conversation folder inside component and make a jsx file and css file for it

Result :



Code :

Conversation.jsx

import './Conversation.css';

function Conversation(props) {

    return (

        <div className='conversation'>

            <img src="https://www.pngall.com/wp-content/uploads/5/Facebook-Messenger-Logo-PNG-High-Quality-Image.png" className='conversationImg' alt="" />

            <span className='conversationName'>Ahmed ben salem</span>

        </div>

    );

}

export default Conversation;

css :

.conversation{

display: flex;

align-items : center ;

padding: 10px;

cursor: pointer;

}

.conversation:hover{

background-color: rgba(189, 188, 188, 0.541);

    }

.conversationImg{

width : 40px ;

height: 40px ;

border-radius: 50%;

object-fit: cover;

margin-right: 20px;

}

.conversationName{

font-weight: 500;

}

Messenger.jsx :

 <div className="chatMenu">

          <div className="chatMenuwrapper">

          <input type="text"

           placeholder="Search for friends"

            className="chatMenuInput" />

            <Conversation />

            <Conversation />

            <Conversation />

            <Conversation />

          </div>

        </div>

* Now time for chatBox :

Make 2 divs inside chatBoxWrapper : chatBoxTop and chatBoxBottom

Make new component named Message and his css file too ;

Message.jsx :

import './message.css';

function Message(props) {

    return (

        <div className='message'>

            <div className="messageTop">

                <img className="msgImg"

                 src="https://www.pngall.com/wp-content/uploads/5/Facebook-Messenger-Logo-PNG-High-Quality-Image.png" alt="" />

                <p className='msgTxt'>

                    Lorem ipsum dolor sit amet consectetur adipisicing elit. Rerum soluta non consequatur eos! Culpa, magnam accusantium! Cumque, vero nemo odio itaque aut non corporis aperiam obcaecati rem nisi! Dignissimos, magni.

                </p>

            </div>

            <div className="messageBottom">

            1 hour ago

            </div>

        </div>

    );

}

export default Message;

message.css :

.message{

    display: flex;

    flex-direction: column;

    margin-top: 20px;

}

.messageTop{

    display: flex;

}

.msgImg{

    width: 32px;

    height: 32px ;

    border-radius: 50%;

    object-fit: cover;

    margin-right: 10px;

}

.msgTxt{

    padding : 10px ;

    border-radius: 20px ;

    background-color: rgb(0, 140, 255);

    color: aliceblue;

    max-width: 300px;

}

.messageBottom{

    font-size: 12px ;

    margin-top : 10px

}

Messenger.jsx :

<div className="chatBox">

          <div className="chatBoxwrapper">

              <div className="chatBoxTop">

            <Message />

            <Message />

            <Message />

            <Message />

              </div>

              <div className="chatBoxBottom">

              </div>

          </div>

        </div>

* Now we gonna work on own msg :

Message.jsx:

        <div className={own ? "message own" : "message"}>

Messenger.jsx :

              <div className="chatBoxTop">

 <Message />

            <Message own={true} />

            <Message />

            <Message own={true} /> …

Message.css :

.message.own{

    align-items: flex-end;

}

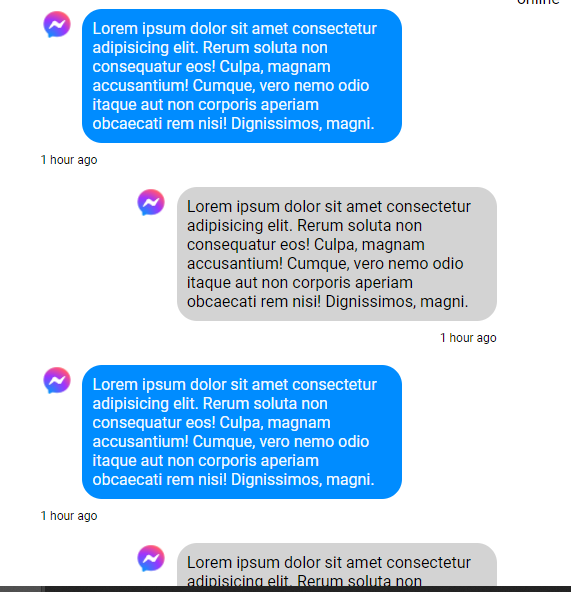
.message.own .msgTxt{

   background-color: lightgray;

   color: black;

}

Result :



* chatBox bottom time :

<div className="chatBoxBottom">

                <textarea  className="chatMsgInput" name="" placeholder="write sthg ..">

                </textarea>

                <button className="chatSubmitButton">Send</button>

              </div>

          </div>

Messeneger.css :

.chatBoxBottom{

     margin-top : 5px ;

     display: flex;

     align-items: center;

     justify-content: space-between;

}

.chatMsgInput{

width: 80%;

height: 90px;

padding: 10px;

margin : 10px 10px

}

.chatSubmitButton{

    width: 70px;height: 40px;border: none;border-radius: 5px;

    cursor: pointer;

    background-color: teal;

    color: wheat;

}

Now we gonna make msgs scrolling :

Add multiple msgs :

 <Message own={true} />

            <Message />

            <Message own={true} />

 <Message own={true} />

            <Message />

            <Message own={true} />

…

Go messenger.css :

*/\* BOX MENU  WRAPPER SIDE \*/*

.chatBoxwrapper{

    display: flex;

    flex-direction: column;

}

// and this will make it scrolling

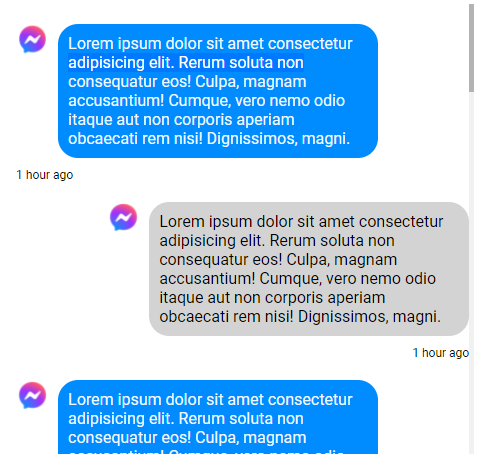
.chatBoxTop{

    height: 100%;

    overflow-y: scroll;

}

Result :



* Time for online section UI but before that let’s take a look about our messeneger.jsx code :

import Conversation from "../../components/Conversations/Conversation";

import Message from "../../components/message/Message";

import Topbar from "../../components/topbar/Topbar";

import "./messenger.css";

function Messenger(props) {

  return (

    <>

      <Topbar />

      <div className="messenger">

        <div className="chatMenu">

          <div className="chatMenuwrapper">

          <input type="text"

           placeholder="Search for friends"

            className="chatMenuInput" />

            <Conversation />

            <Conversation />

            <Conversation />

            <Conversation />

          </div>

        </div>

        <div className="chatBox">

          <div className="chatBoxwrapper">

              <div className="chatBoxTop">

            <Message />

            <Message own={true} />

            <Message />

            <Message own={true} />

            <Message own={true} />

            <Message />

            <Message own={true} /> <Message own={true} />

            <Message />

            <Message own={true} /> <Message own={true} />

            <Message />

            <Message own={true} />

              </div>

              <div className="chatBoxBottom">

                <textarea className="chatMsgInput" name="" placeholder="write sthg ..">

                </textarea>

                <button className="chatSubmitButton">Send</button>

              </div>

          </div>

        </div>

        <div className="chatOnline">

          <div className="chatOnlinewrapper">

              online

          </div>

        </div>

      </div>

    </>

  );

}

export default Messenger;

Ok, let’s attack ChatOnline div :

1. Make new component named ChatOnline and his css file :

chatOnline Jsx file :

import './chatOnline.css';

function ChatOnline(props) {

    return (

        <div className='chatOnline'>

            <div className="chatOnlineFriend">

            </div>

            <div className="chatOnlineImgContainer">

                <img  className='chatOnlineImg'

                src="https://www.pngall.com/wp-content/uploads/5/Facebook-Messenger-Logo-PNG-High-Quality-Image.png" alt="" />

                <div className="chatOnlineBadge">

                </div>

                <span className="chatOnlineName">

                    Sofien marrouki

                </span>

            </div>

        </div>

    );

}

export default ChatOnline;

chatOnline.css :

.chatOnlineImgContainer{

display: flex;

align-items: center;

position: relative;

margin-right:  10px;

}

.chatOnlineFriend{

    margin-top: 10px;

    display: flex;

    align-items: center;

    font-weight: 500;

    cursor : pointer

}

.chatOnlineImg{

    width: 32px;

    height: 32x;

    border-radius: 50%;

    object-fit: cover;

    border: 1px solid wheat;

}

.chatOnlineBadge{

    position: absolute;

    width: 10px;

    border-radius: 50%;

    background-color: rgb(31, 255, 31);

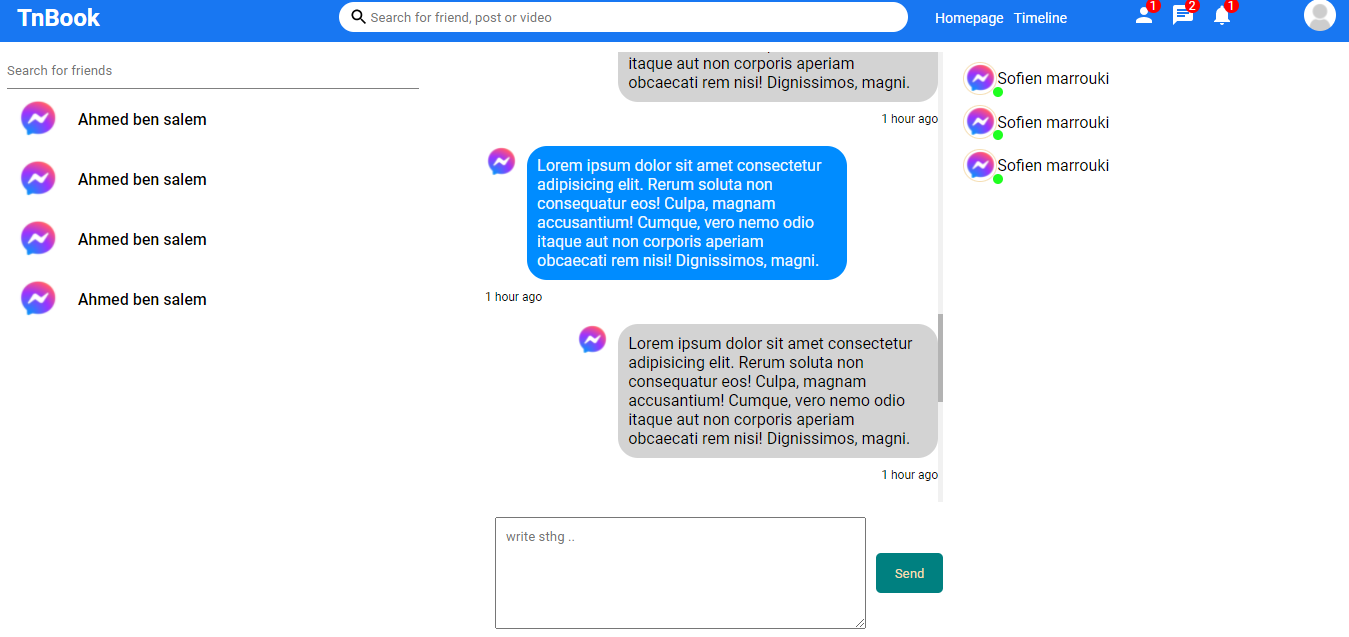
    height: 10px;

    top : 25px ;

   left: 30px;

}

Res :



1. REST API :
2. Create 2 models : conversation.js and message.js

const mongoose = require("mongoose");

const ConversationSchema = new mongoose.Schema(

  {

    members: {

      type: Array,

    },

  },

  { timestamps: true }

);

module.exports = mongoose.model("Conversation", ConversationSchema);

Message model :

const mongoose = require("mongoose");

const MessageSchema = new mongoose.Schema(

  {

    conversationId: {

      type: String,

    },

    sender :{

        type:String

    },

    text:{

        type : String

    }

  },

  { timestamps: true }

);

module.exports = mongoose.model("Message", MessageSchema)

1. Create 2 routes , one for messages and one for conversations :
2. Conversations.js

const router = require("express").Router();

const Conversation = require("../models/Conversation");

*//new conv*

router.post("/", async (req,res) => {

    const newConversation = new Conversation({

        members : [req.body.senderId, req.body.reciverId]

    })

    try {

        const savedConversation = await newConversation.save();

        res.status(200).json(savedConversation)

    } catch (error) {

        res.status(500).json(error)

    }

})

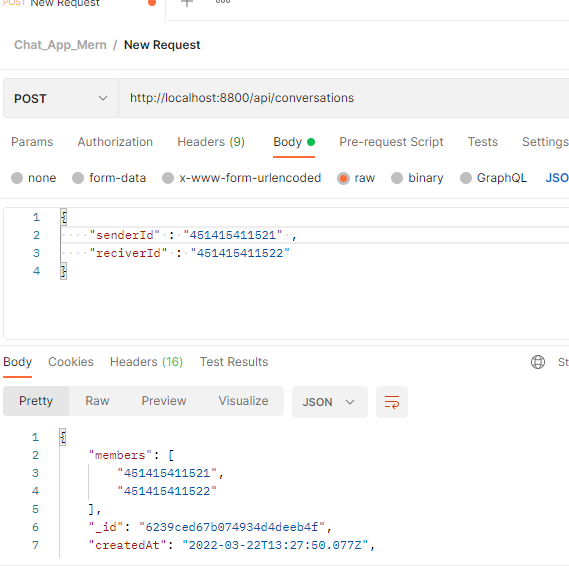
*//get conv of a user*

module.exports = router;

import route inside index.js :

app.use("/api/conversation" ,conversationsRoute )

and test the request with postman :



* GET USER CONVERSATION :

*//get conv of a user*

router.get("/:userId", async (req,res)=>{

    try {

        const conversation = await Conversation.find({

            members : { $in:[req.params.userId] }

        })

        res.status(200).json(conversation)

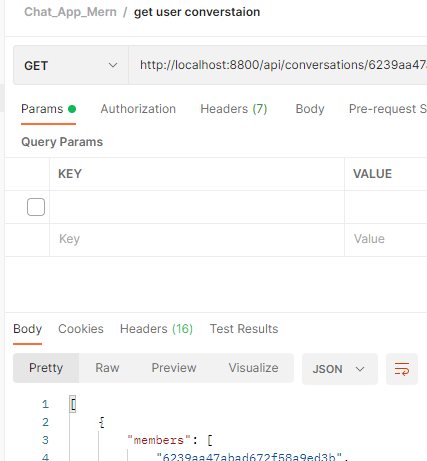
    } catch (error) {

        res.status(500).json(error)

    }

})

Test it with postman :



Now we do same work for messages , creation of routes and import them , test requests ..

1. Messages.js :

const router = require("express").Router();

const Message = require("../models/Message");

*//add*

router.post("/",async (req, res) => {

    const newMsg = new Message(

        req.body

    )

    try {

        const saveMsg = await newMsg.save()

        res.status(200).json(saveMsg)

    } catch (error) {

        res.status(500).json(error)

    }

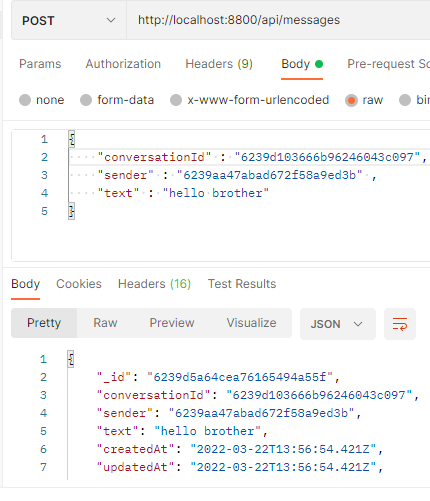
})

*// get*

module.exports = router;

import it inside index.js for messagesRoutes .

Test post a new msg with postman :



*// get*

router.get("/:conversationId",async (req, res) => {

    try {

        const msgs = await Message.find(

            {

                conversationId:req.params.conversationId

            }

        )

        res.status(200).json(msgs)

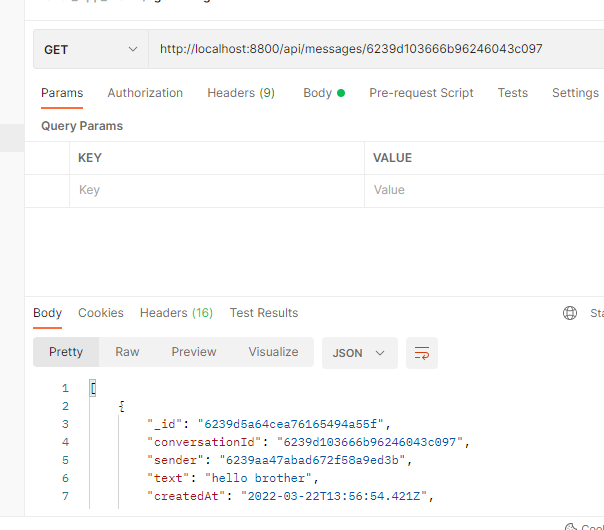
    } catch (error) {

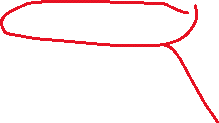
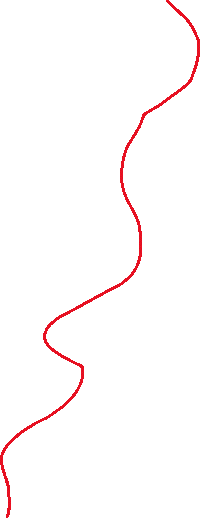
        res.status(500).json(error)

    }

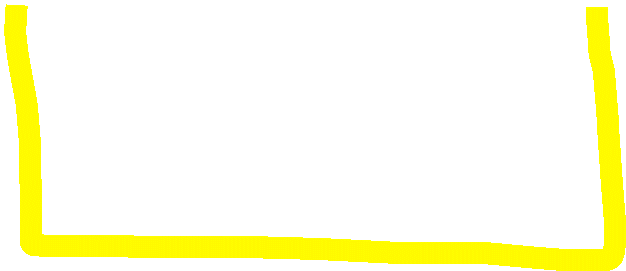
})

Test get msg with post man :









1. FETCHING USER CONVERSATION FROM DB :

<<Return to client side >>

http://localhost:3000/messenger

function Messenger(props) {

  const {user} = useContext(AuthContext);

  console.log(user)

 useEffect(() => {

const getConversations = async () =>{

try {

  const res = await axios.get('/conversations/'+user.\_id)

*// console.log(res)*

  setConversations(res.data);

} catch (error) {

  console.log(error)

}

}

getConversations();

  },[user.\_id])

No more static display :

  <input type="text"

           placeholder="Search for friends"

            className="chatMenuInput" />

            {

              console.log(conversations)

            }

            {conversations && conversations.map( c =>

            <Conversation key={c.\_id} conversation={c} currentUser={user} />

            )

            }

            {*/\* <Conversation />*

*<Conversation />*

*<Conversation />*

*<Conversation /> \*/*}

Conversation.jsx :

First of all follow a second user with users routes and postman help .

And make a conversation between them including msgs .

import axios from 'axios';

import { useEffect, useState } from 'react';

import './Conversation.css';

function Conversation({conversation , currentUser}) {

    const [user,setUser] = useState(null);

    useEffect(()=>{

    const friendId = conversation.members.

    find( m => m !== currentUser.\_id)

    const getUser = async () => {

try {

    const res = await axios.get("/users?userId="+friendId)

*// console.log("res data : ",res.data)*

    setUser(res.data)

} catch (error) {

    console.log(error)

}

    };

getUser();

    },[currentUser,conversation])

    return (

        <div className='conversation'>

            <img src={user && user.profilePicture !== '' ? user.profilePicture : "https://www.pngall.com/wp-content/uploads/5/Facebook-Messenger-Logo-PNG-High-Quality-Image.png"} className='conversationImg' alt="" />

            <span className='conversationName'>{ user && user.username }</span>

        </div>

    );

}

export default Conversation;

1. FETCHING MESSAGES FROM DB:

At Messanger page , Messenger.jsx :

  const [conversations,setConversations] = useState([]);

  const [currentChat,setcurrentChat] = useState(null);

  const [messages,setMessages] = useState([]);

 <div className="chatBoxwrapper">

            {currentChat ?

            <>

              <div className="chatBoxTop">

            <Message />

            <Message own={true} />

              </div>

              <div className="chatBoxBottom">

       <textarea className="chatMsgInput" name="" placeholder="write sthg ..">

                </textarea>

                <button className="chatSubmitButton">Send</button>

              </div> </>

: <span className="NoConversationText"> open conversation to start chat  ..</span>}

Css of .NoConversationText :

.NoConversationText{

    display: flex;

    justify-content: center;

    margin-top: 50%;

    background-color: rgba(241, 241, 241, 0.151);

    text-decoration: underline;

}

setCurrentChat :

  {conversations && conversations.map( c =>

            <div key={c.\_id} onClick={()=>setCurrentChat(c)}>

            <Conversation  conversation={c} currentUser={user} />

            </div>

            )

            }

Messenger.jsx :

useEffect(()=>{

  const getMessages = async () => {

   try {

    const res = await axios.get("/messages/" + currentChat?.\_id);

    setMessages(res.data)

   } catch (error) {

     console.log(error)

   }

  };

  getMessages()

},[currentChat])

*// console.log(messages)*

Display messages :

 <div className="chatBoxwrapper">

            {currentChat ?

            <>

              <div className="chatBoxTop">

                {  messages.map( (m) => (

                     <Message  key={m.\_id} message={m} own={m.sender === user.\_id} />

                ))}

              </div>

              <div className="chatBoxBottom">

                <textarea className="chatMsgInput" name="" placeholder="write sthg ..">

                </textarea>

                <button className="chatSubmitButton">Send</button>

              </div> </>

               : <span className="NoConversationText"> open conversation to start chat  ..</span>}

          </div>

Message.jsx will get props and display it so :

import './message.css';

import {format} from 'timeago.js'

function Message({message , own}) {

    console.log(message)

    return (

        <div className={own ? "message own" : "message"}>

            <div className="messageTop">

                <img className="msgImg"

                 src="https://www.pngall.com/wp-content/uploads/5/Facebook-Messenger-Logo-PNG-High-Quality-Image.png" alt="" />

                <p className='msgTxt'>

                    {message ? message.text : "no msg here" }

                </p>

            </div>

            <div className="messageBottom">

            {message ? format(message.createdAt): " delivred "}

            </div>

        </div>

    );

}

export default Message;

TopBar.jsx :

  <Link to="/messenger" >

          <div className="topbarIconItem">

            <Chat />

            <span className="topbarIconBadge">2</span>

          </div>

          </Link>

1. POST NEW CHAT / REACT SCROLL TO END OF DIV:
2. SOCKET.IO SEND MSGS:
3. END: