Messanger-Clone (chat app)

Step1 → set up firebase. {

Step2 → Do file clean up.

Step3 → create a input, and a state for input.

Step4 → create a button, onClick, sendmessage func^n and a state for storing msgs.

Step5 → write a map func^n, and map through all the messages from messages array.

Step6 → wrap the input and btn inside a form, so that we can have enter functionality, don't forget to do event.preventDefault();.

Step7 → change normal button to material-ui button, also apply disabled = {! input}, so that it don't take empty values.

Step8 → Put input & btn inside formcontrol of material-ui.

Step9 → Create a message Component.

Step10 → Create a Username state.

Step11 → Create a useEffect which will fire once when App component loads, and in the cond^n part, display a prompt on screen and get the username.

Step12 → Convert normal array into array of objects, where each object

has username and text, and pass them accordingly as props to message component.
- also remember to ~~update~~ change the spread operator code.

Step 13 → In messages comp, display a message in this manner -

Rahul : Hello there
↑ ↑
username text

1:00 hr

Step 14 → Let's design messages using Material UI.
In message component, wrap the content inside card from Material UI, give the card a className = "message ~~card~~"

Step 15 → Create a message.css.

Step 16 → change the ~~attributes~~ of Message comp. in App.js, b'coz now we want to add the functionality that if the user who is logged in is sending the message, then the msg should be on right side with blue background.

Step 17 → Setup database in firebase, name the collection as messages.

Step 18 → Do npm install firebase and create a firebase.js file.

Step 19 → create useEffect in App.js & setMessages by taking object from the database.

Step 20 → In the ~~useEffect that was created earlier~~ sendMessage func^n in app.js, in that write the code for adding the message data into firebase database.

Step 21 → write a .orderBy ('timestamp', 'desc') before onSnapshots

Step 22 → To give a little bit animation to messages, add Flip move.

Step 23 → Also, we need that whenever we add a new msg, our whole list should not get refreshed, for that add key to the message component.

2:00 hr

Step 24 → Add messanger image at the top.

Step 25 → Bring send message thing at the bottom of the page.
In App.js, Give the form tag, a className = "app--form". and add css to it.

Step 26 → Add a send icon from material-ui instead of the send Message button.
To make the send icon, as a button, import <IconButton> from material ui, & wrap the icon inside <IconButton>.

```
<IconButton>
    <SendIcon/>
</IconButton>
```

step 27 For the user who is logged in and sending the msg don't display his name, and show names for the other msgs.

step 28. Give the FormControl, a className= app--formControl and add css to it. {display: flex
flex-direction: row

step 29 Give the IconButton, a Class name= app--iconButton and add css to it. Flex: 0

Give Input a className = app--input, and add css to it. flex: 1

Step 30 Now, when the user is not there, at the place of Username it is showing null but we don't want that, instead of null it should say unknown user. For that go to message.js.

```
{!isUser && `${message.username || 'unknown U
  `} {message.message}
```

step 31 deploy it to firebase.