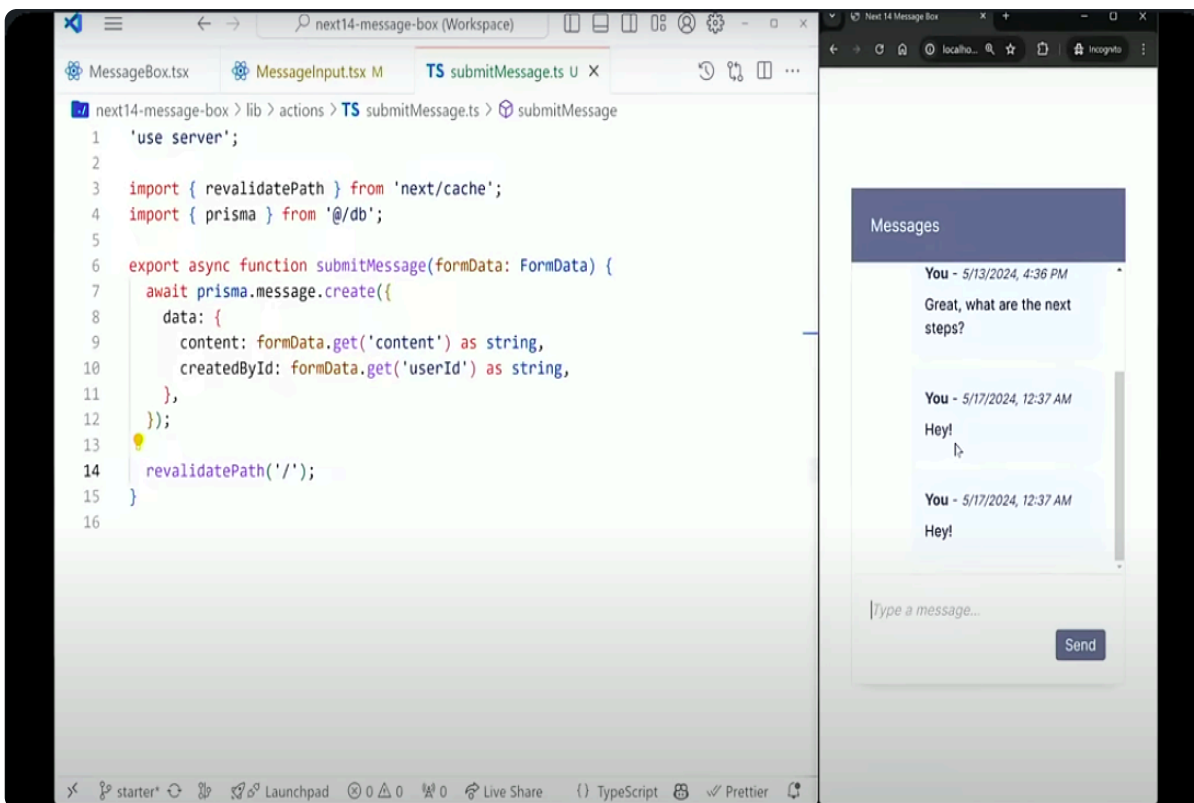
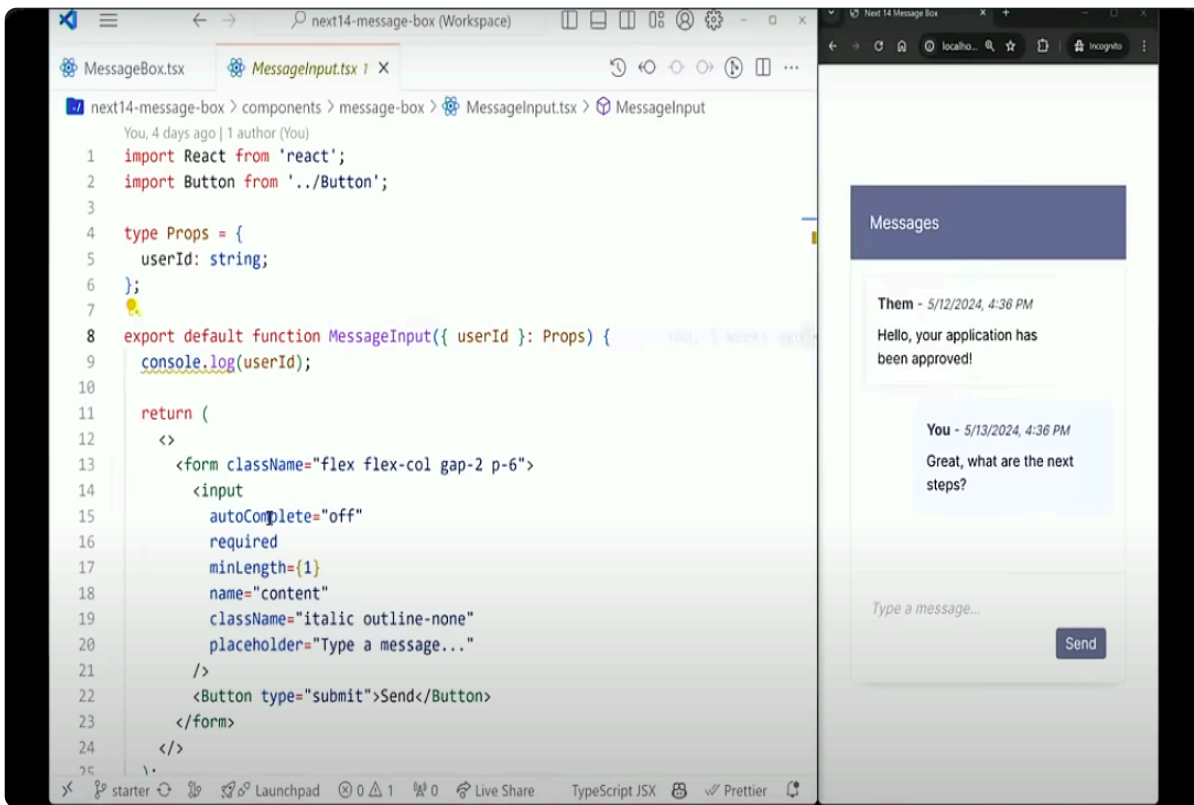


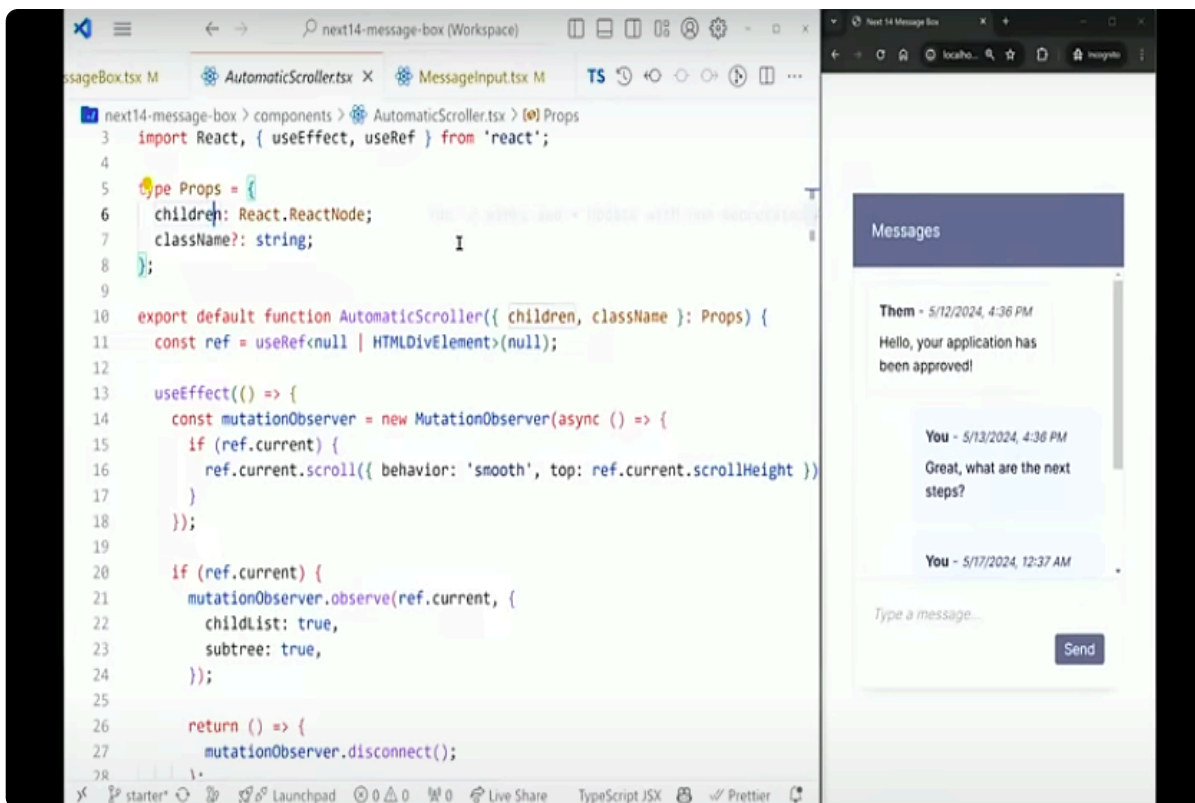
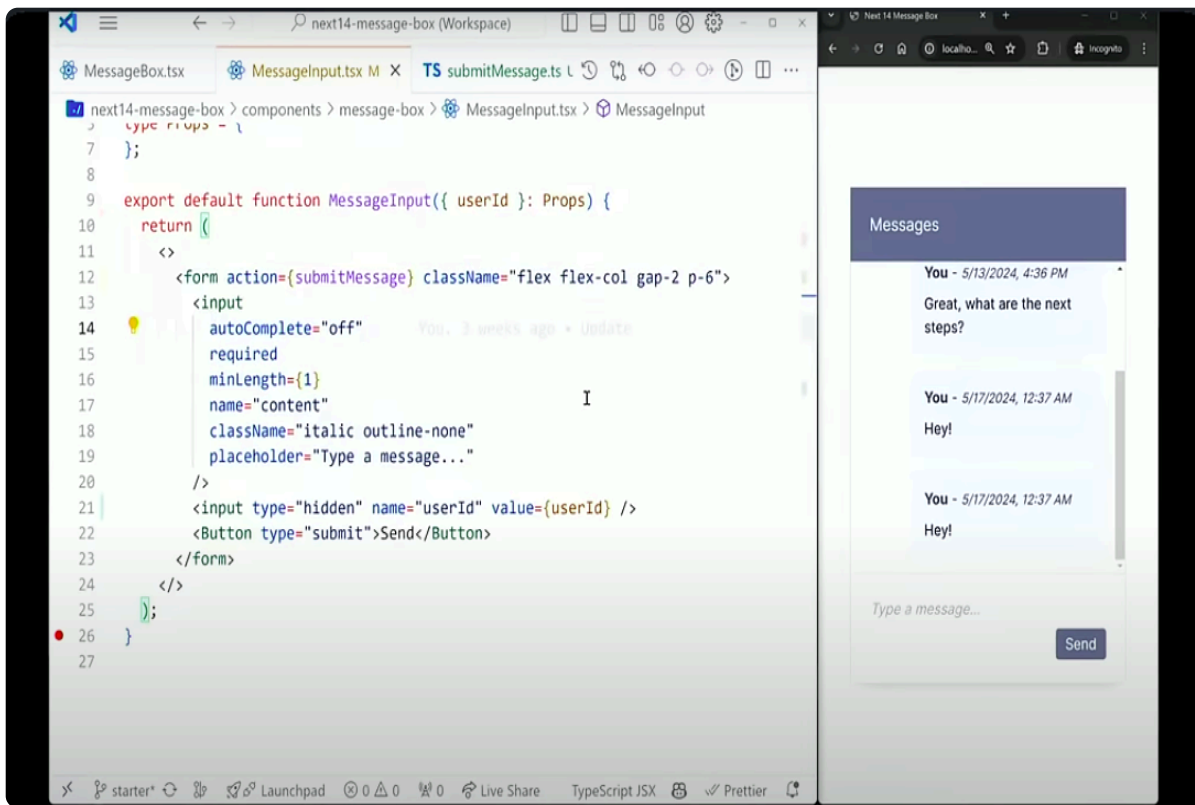


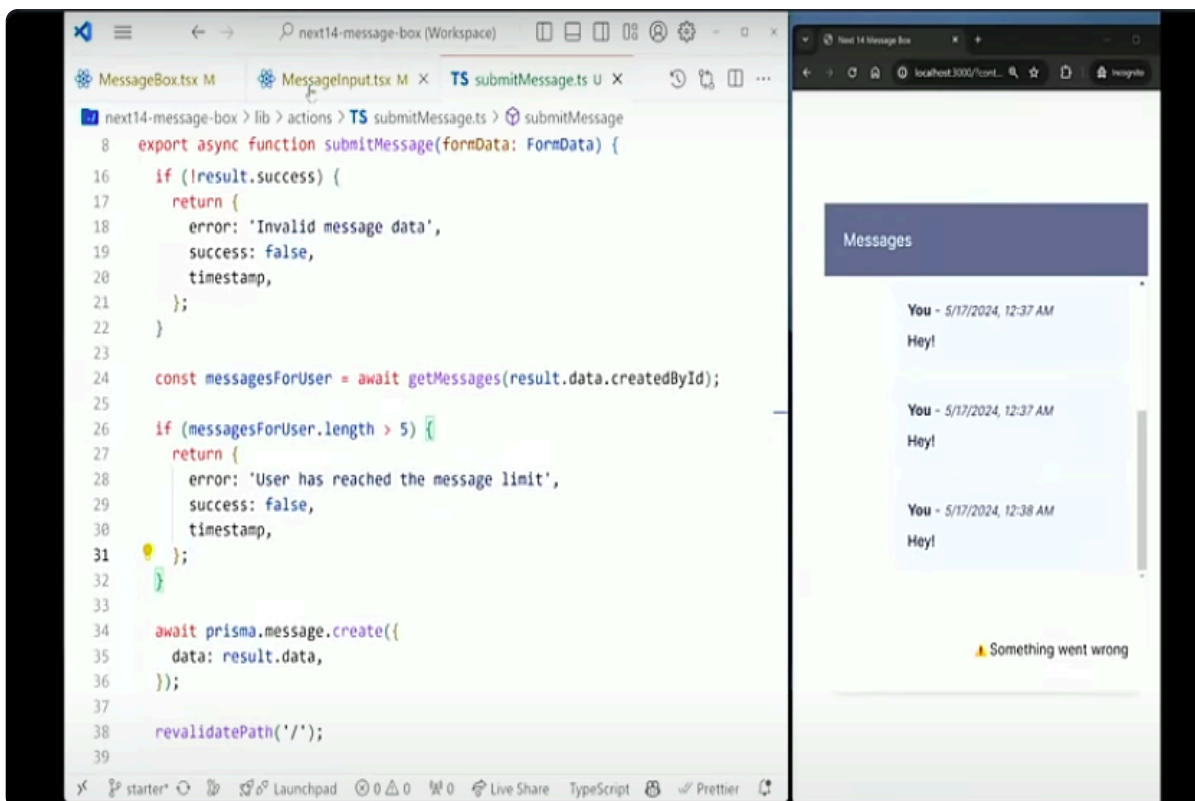
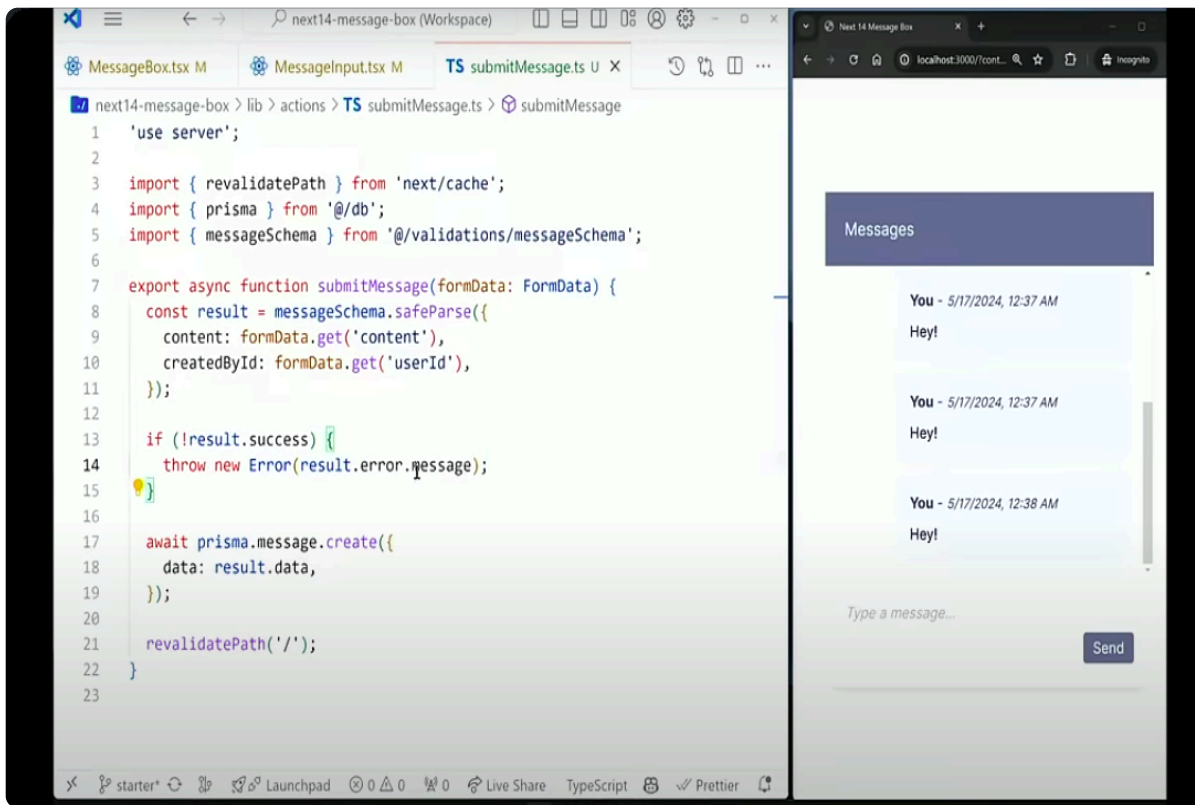
# Enhancing Forms with RSC

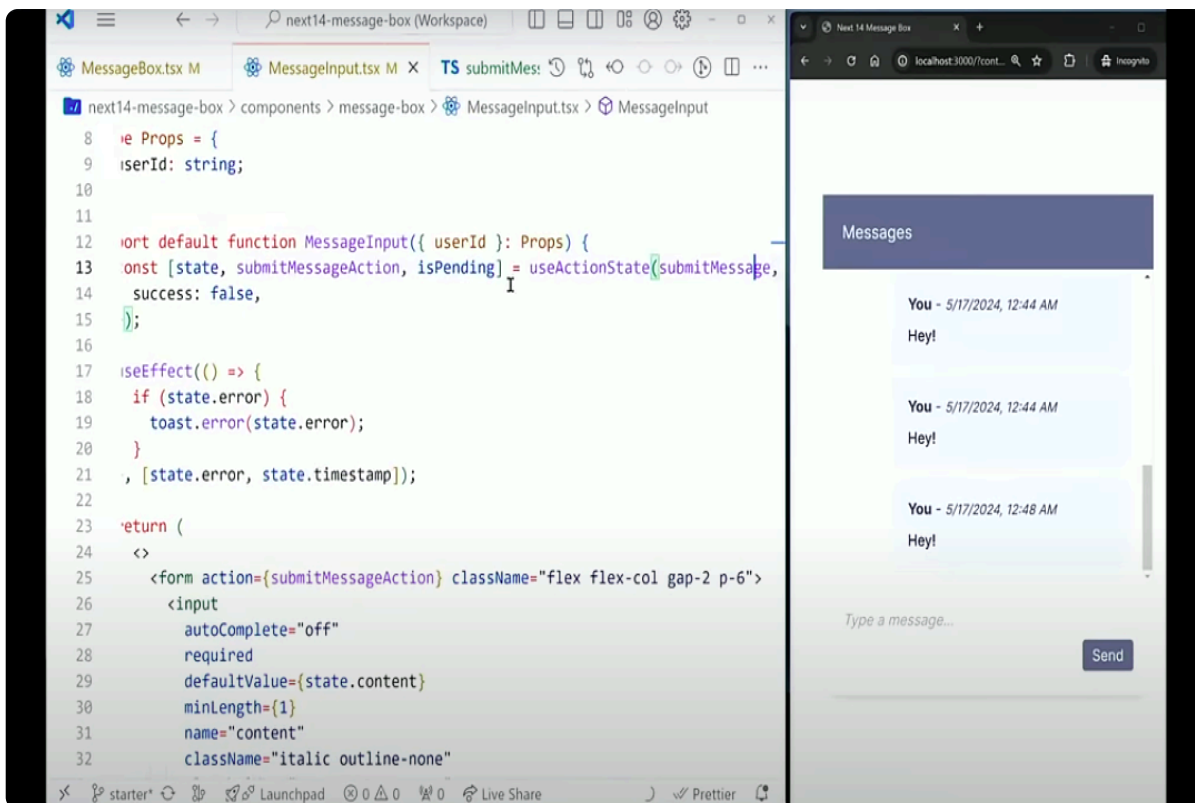
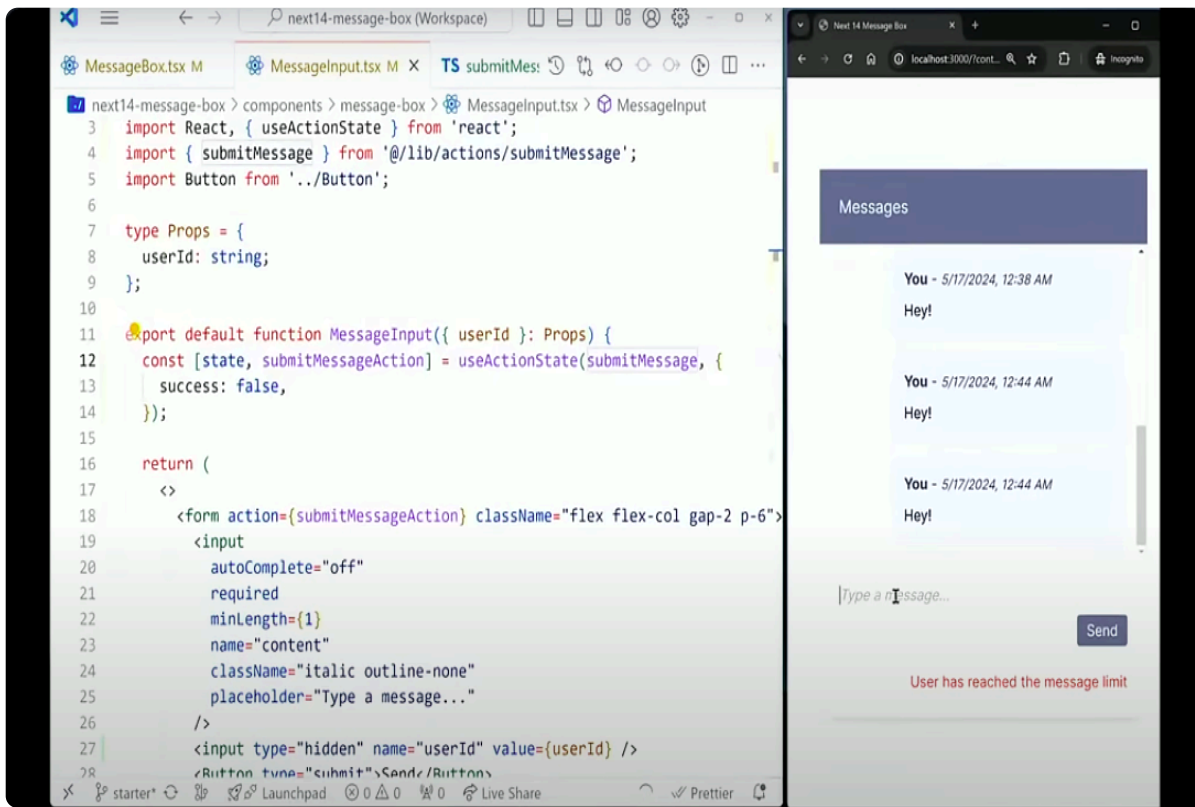
```
MessageBox.tsx
next14-message-box > components > message-box > MessageBox.tsx > ...
1 import React from 'react';
2 import { getCurrentUser } from '@lib/services/getCurrentUser';
3 import { getMessages } from '@lib/services/getMessages';
4 import MessageDisplay from './MessageDisplay';
5 import MessageInput from './MessageInput';
6
7 export default async function MessageBox() {
8   const messages = await getMessages();
9   const user = await getCurrentUser();
10
11   return (
12     <div className="flex w-full flex-col shadow-xl sm:w-[400px]">
13       <h1 className="bg-slate-500 p-6 text-lg text-white">Messages</h1>
14       <div className="grid border-x border-b border-gray-300">
15         <div className="grid h-80 content-start gap-4 overflow-auto border-b">
16           {messages.length === 0 && <span className="text-center text-gray-500">No messages yet</span>}
17           {messages.map(message => {
18             return <MessageDisplay userId={user.id} key={message.id} message={message} />
19           })}
20         </div>
21         <MessageInput userId={user.id} />
22       </div>
23     </div>
24   );
25 }
```

```
MessageDisplay.tsx
next14-message-box > components > message-box > MessageDisplay.tsx > MessageDisplay
5 type Props = {
6   userId: string;
7 };
8
9
10 export default function MessageDisplay({ message, userId }: Props) {
11   const isWrittenByUser = userId === message.createdById;
12
13   return (
14     <div
15       className={cn(
16         'flex h-fit w-4/5 flex-col gap-2 rounded p-4 sm:w-2/3',
17         isWrittenByUser ? 'justify-self-end bg-slate-200' : 'justify-self-start'
18       )}
19     >
20       <span className="text-slate-700">
21         <span className="font-bold">{isWrittenByUser ? 'You' : 'Them'}</span>
22         { ' - ' }
23         <span className="text-nowrap text-sm italic">
24           {message.createdAt.toLocaleString('en-US', {
25             day: 'numeric',
26             hour: 'numeric',
27             minute: 'numeric',
28             month: 'numeric',
29             year: 'numeric',
30           })}
31         </span>
32       </span>
33     </div>
34   );
35 }
```









```

12  };
13
14  type Props = {
15    messages: Message[];
16    userId: string;
17  };
18
19  export default function Messages({ messages, userId }: Props) {
20    const [optimisticMessages, addOptimisticMessage] = useOptimistic(
21      messages,
22      (currMessages: OptimisticMessage[], newMessage: OptimisticMessage) => {
23        return [
24          ...currMessages,
25          {
26            ...newMessage,
27            isSending: true,
28          },
29        ];
30      },
31    );
32
33    return (
34      <div className="grid border-x border-b border-gray-300">
35        <AutomaticScroller className="grid h-80 content-start gap-4 overflow-auto">
36          {optimisticMessages.length === 0 && <span className="text-center">No messages</span>}
37          {optimisticMessages.map(message => {

```

