KHUSHI KALRA

22CS3035

LAB-08

T1

```jsx
code > src > ⚛ App.jsx > …
1    import React, { useState } from 'react';
2
3    function App() {
4        const [usdAmount, setUsdAmount] = useState('');
5        const [eurAmount, setEurAmount] = useState('');
6        const usdToEurRate = 0.85;
7        const handleUsdChange = (event) => {
8            setUsdAmount(event.target.value);
9            setEurAmount(event.target.value * usdToEurRate);
10       }
11       return (
12           <div>
13               <h1>Currency Converter</h1>
14               <div>
15                   <label>USD: </label>
16                   <input
17                       type="number"
18                       value={usdAmount}
19                       onChange={handleUsdChange}
20                   />
21               </div>
22               <div>
23                   <label>EUR: </label>
24                   <input type="text" value={eurAmount} readOnly />
25               </div>
26           </div>
27       )
28   }
29
30   export default App;
```

Currency Converter

USD: [   ]
EUR: [   ]

Currency Converter

USD: [8 ▲▼]
EUR: [5.1]

Currency Converter

USD: [67 ▲▼]
EUR: [56.949999999999996]

T2

```
code > src > ⚛ App.jsx > ...
1    import React, { useState } from 'react';
2
3    function App() {
4      const [isRunning, setIsRunning] = useState(false);
5      const [timer, setTimer] = useState(0);
6
7      const startTimer = () => {
8        setIsRunning(true);
9        const interval = setInterval(() => {
10         setTimer((prevTimer) => prevTimer + 1);
11       }, 1000);
12       return () => clearInterval(interval);
13     };
14     const pauseTimer = () => {
15       setIsRunning(false);
16     };
17     const resetTimer = () => {
18       setIsRunning(false);
19       setTimer(0);
20     };
21     return (
22       <div className="container">
23         <h1>Stopwatch</h1>
24         <p className="timer">Timer: {timer}</p>
25         <button className="btn" onClick={isRunning ? pauseTimer : startTimer}>
26           {isRunning ? 'Pause' : 'Start'}
27         </button>
28         <button className="btn" onClick={resetTimer}>Reset</button>
29       </div>
30     );
31   }
32
33   export default App;
```

Stopwatch

Timer: 10

Pause   Reset

Stopwatch

Timer: 22

Pause   Reset

Stopwatch

Timer: 0

Start   Reset

T3

```jsx
1   import React, { useState } from 'react';
2
3   const MessagingApp = () => {
4     const [conversations, setConversations] = useState([
5       { id: 1, name: 'Family', messages: [] },
6       { id: 2, name: 'Friends', messages: [] },
7     ]);
8     const [selectedConversation, setSelectedConversation] = useState(null);
9     const [newMessage, setNewMessage] = useState('');
10
11    const handleConversationClick = (conversationId) => {
12      setSelectedConversation(conversationId);
13    };
14
15    const handleSendMessage = () => {
16      if (newMessage.trim() !== '' && selectedConversation) {
17        setConversations((prevConversations) =>
18          prevConversations.map((conversation) =>
19            conversation.id === selectedConversation
20              ? {
21                  ...conversation,
22                  messages: [
23                    { text: newMessage, timestamp: new Date().toLocaleTimeString() },
24                    ...conversation.messages,
25                  ],
26                }
27              : conversation
28          )
29        );
30
31        setNewMessage('');
32      }
33    };
```

```jsx
35    return (
36      <div>
37        <h1>Messaging App</h1>
38
39        <div style={{ display: 'flex' }}>
40          <div style={{ flex: '1', borderRight: '1px solid #ccc', padding: '10px' }}>
41            <h2>Conversations</h2>
42            <ul>
43              {conversations.map((conversation) => (
44                <li key={conversation.id} onClick={() => handleConversationClick(conversation.id)}>
45                  {conversation.name}
46                </li>
47              ))}
48            </ul>
49          </div>
50          <div style={{ flex: '3', padding: '10px' }}>
51            {selectedConversation && (
52              <div>
53                <h2>{`Chat with ${conversations.find((conv) => conv.id === selectedConversation).name}`}</h2>
54                <div style={{ maxHeight: '300px', overflowY: 'auto', border: '1px solid #ccc', padding: '10px' }}>
55                  {conversations
56                    .find((conv) => conv.id === selectedConversation)
57                    .messages.map((message, index) => (
58                      <div key={index} style={{ borderBottom: '1px solid #eee', paddingBottom: '5px' }}>
59                        <strong>{message.timestamp}</strong>: {message.text}
60                      </div>
61                    ))}
62                </div>
63                <textarea
64                  rows="3"
65                  value={newMessage}
66                  onChange={(e) => setNewMessage(e.target.value)}
67                />
68                <button onClick={handleSendMessage}>Send</button>
69              </div>
70            )}
71          </div>
72        </div>
73      </div>
74    );
75  };
76
77  export default MessagingApp;
```

Messaging App

Conversations

- Family
- Friends

Chat with Friends

[   ]

[   ]
[   ] Send