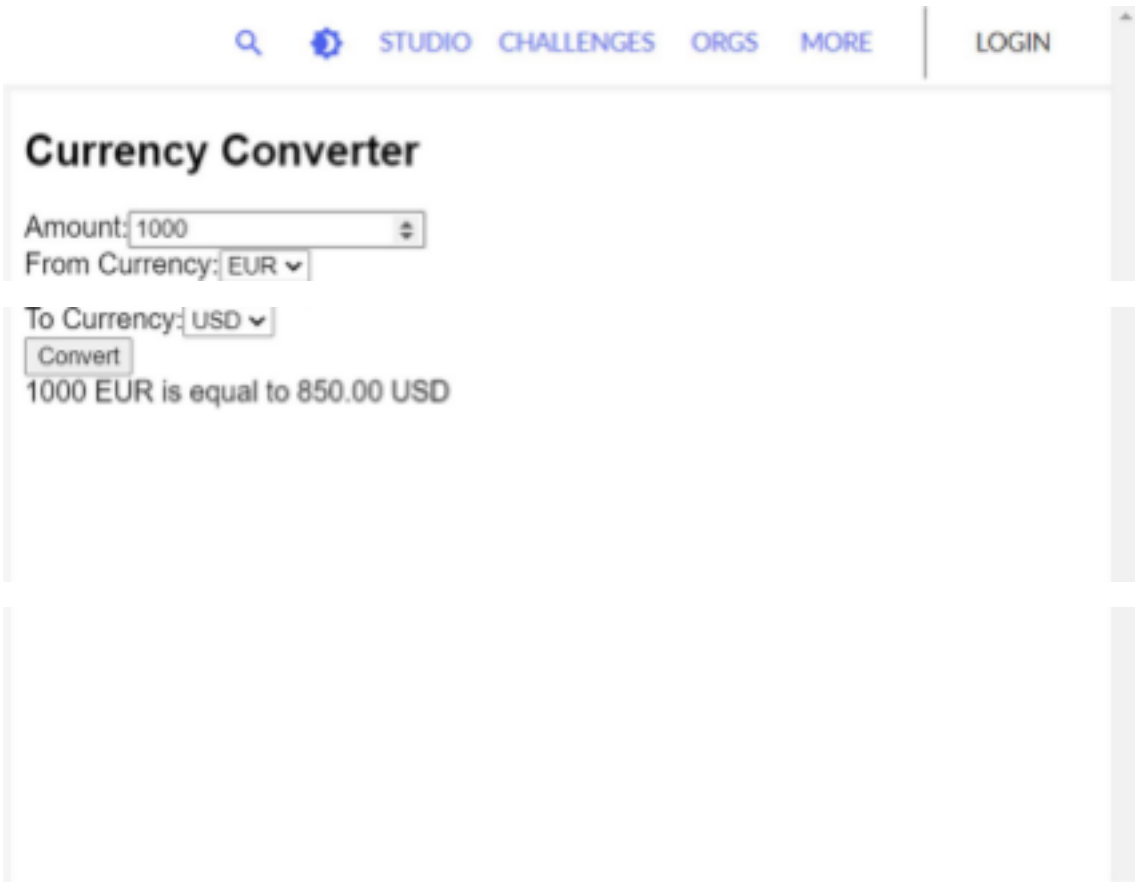
CODE-1

```
import React, { useState } from 'react';

funcon CurrencyConverter() {
 const [amount, setAmount] = useState(0);
 const [fromCurrency, setFromCurrency] =
useState('USD'); const [toCurrency, setToCurrency] =
useState('EUR'); const [convertedAmount,
setConvertedAmount] = useState(0);

 // Hard-coded exchange rate
 const exchangeRate = 0.85; // 1 USD = 0.85 EUR
 const handleAmountChange = (e) => {
 const value = parseFloat(e.target.value);
 setAmount(value);
 };

 const handleFromCurrencyChange = (e) => {
 setFromCurrency(e.target.value);
 };

 const handleToCurrencyChange = (e) => {
```

```
setToCurrency(e.target.value);

};


const convertCurrency = () => {

const result = amount * exchangeRate;

setConvertedAmount(result.toFixed(2)); // Round to 2
decimal places

};


return (

<div>

<h2>Currency Converter</h2>

<div>
<label>

Amount:

<input type="number" value={amount}
onChange={handleAmountChange} />
</label>

</div>

<div>

<label>

From Currency:

<select value={fromCurrency}
onChange={handleFromCurrencyChange}>

<opon value="USD">USD</opon> <opon

value="EUR">EUR</opon> </select>

</label>
```

```
 </div>
 <div>
 <label>
 To Currency:
 <select value={toCurrency}
onChange={handleToCurrencyChange}>
<opon value="USD">USD</opon>
<opon value="EUR">EUR</opon>
</select>
 </label>
 </div>
 <buon onClick={convertCurrency}>Convert</buon>
<div>
 {amount} {fromCurrency} is equal to {convertedAmount}
{toCurrency}
 </div>
 </div>
 );
}

export default CurrencyConverter;
```

OUTPUT



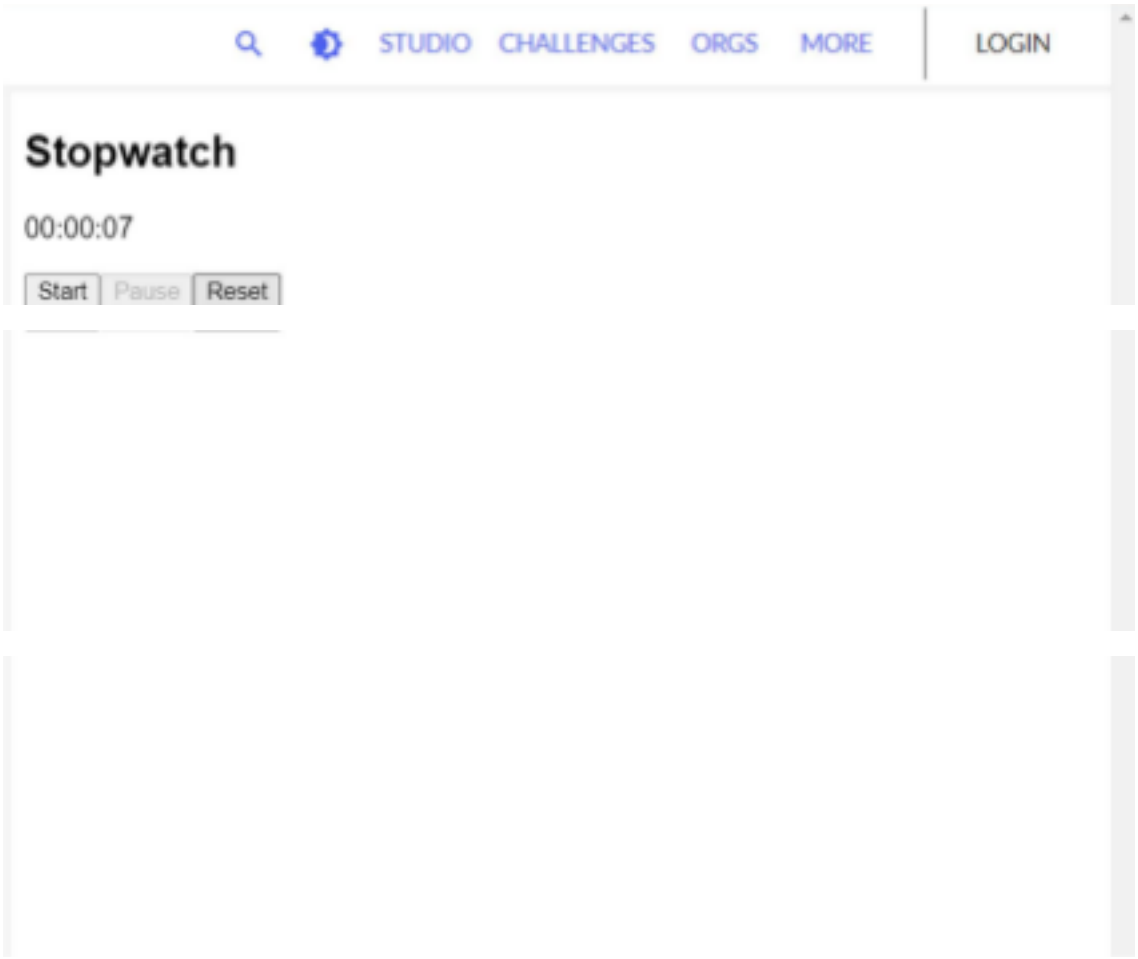
CODE-2

```
import React, { useState, useEffect } from 'react';

funcon Stopwatch() {
 const [mer, setTimer] = useState(0); const
[isRunning, setIsRunning] = useState(false);

 useEffect(() => {
 let intervalId;

 if (isRunning) {
 intervalId = setInterval(() => {
 setTimer((prevTimer) => prevTimer + 1);
```

```
}, 1000);
 } else {
 clearInterval(intervalId);
 }
 return () =>
clearInterval(intervalId); },
[isRunning]);

 const startTimer = () => {
 setIsRunning(true);
 };

 const pauseTimer = () => {
 setIsRunning(false);
 };

 const resetTimer = () => {
 setTimer(0);
 setIsRunning(false);
 };

 const formatTime = (me) => {
 const hours = Math.floor(me / 3600);
const minutes = Math.floor((me % 3600) /
60); const seconds = me % 60;
 return `${hours.toString().padStart(2,
```

```
'0')}:${minutes .toString()
 .padStart(2, '0')}:${seconds.toString().padStart(2,
'0')}`; };

 return (
 <div>
 <h2>Stopwatch</h2>
 <div>
 <p>{formatTime(mer)}</p>
 </div>
 <div>
 <buon onClick={startTimer} disabled={isRunning}>
Start
 </buon>
 <buon onClick={pauseTimer} disabled={!isRunning}>
Pause
 </buon>
 <buon onClick={resetTimer}>Reset</buon>
</div>
 </div>
 );
}

export default Stopwatch;
```

OUTPUT



CODE-3

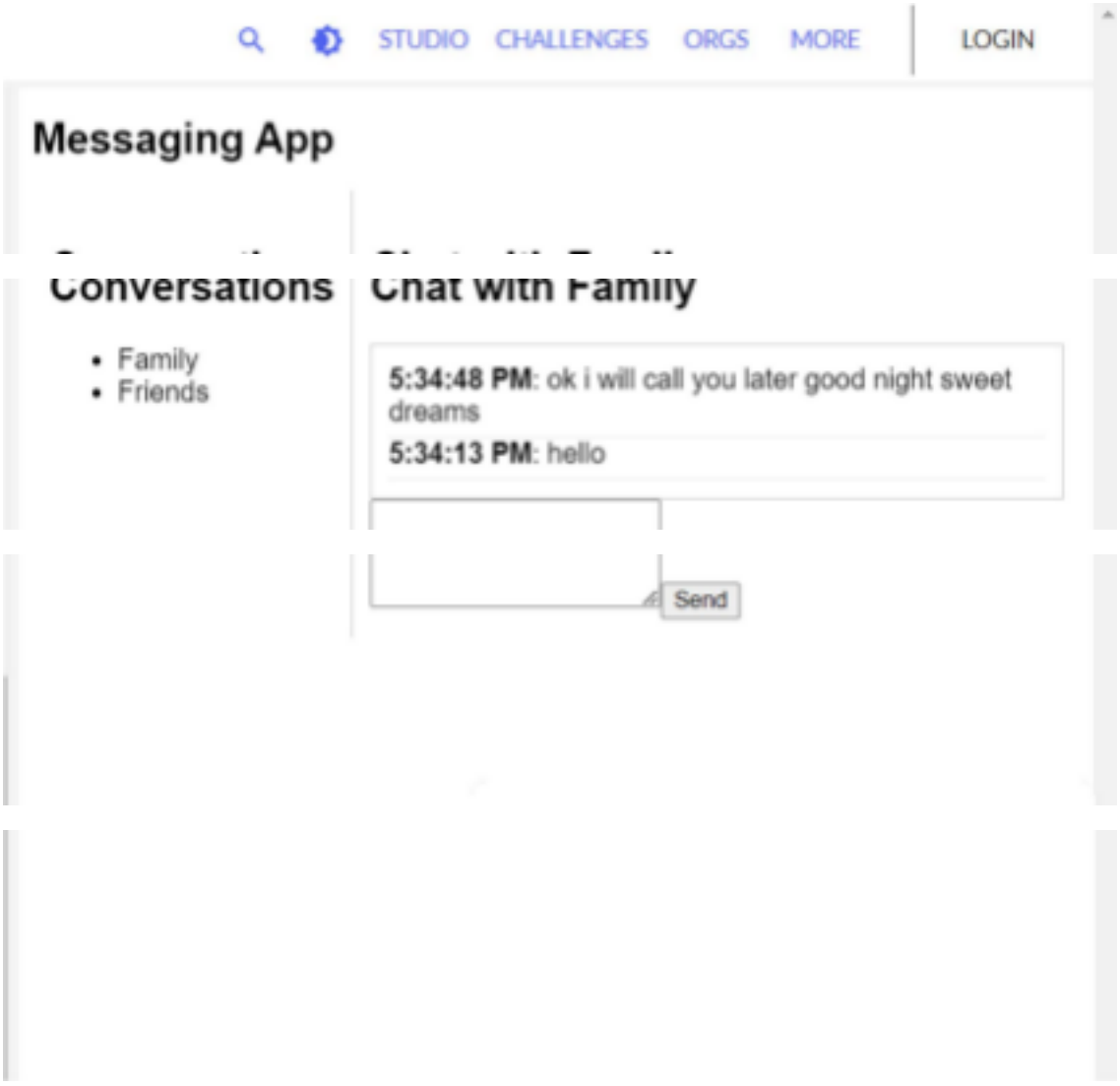
```
import React, { useState } from 'react';
const MessagingApp = () => {

 const [conversaons, setConversaons] =

useState([ { id: 1, name: 'Family', messages: [] },

 { id: 2, name: 'Friends', messages: [] },

 ]);

 const [selectedConversaon,
setSelectedConversaon] = useState(null);
```

```
const [newMessage, setNewMessage] = useState('');
const handleConversaonClick = (conversaonId)
=> { setSelectedConversaon(conversaonId);
};

const handleSendMessage = () => {
if (newMessage.trim() !== '' && selectedConversaon) {
const updatedConversaons = conversaons.map((conversaon)
=>
conversaon.id === selectedConversaon
? {
...conversaon,
messages: [
{ text: newMessage, mestamp: new
Date().toLocaleTimeString() },
...conversaon.messages,
],
}
: conversaon
);
setConversaons(updatedConversaons);
setNewMessage('');
}
};
return (
<div>
<h1>Messaging App</h1>
```

```
 <div style={{ display: 'flex' }}>
 {/* List of Conversaons */}
 <div style={{ flex: '1', borderRight: '1px solid #ccc',
padding: '10px' }}>
 <h2>Conversaons</h2>
 <ul>
 {conversaons.map((conversaon) => (
 <li key={conversaon.id} onClick={() =>
handleConversaonClick(conversaon.id)}>
 {conversaon.name}
 </li>
 ))}
 </ul>
 </div>

 {/* Chat Interface */}
 <div style={{ flex: '3', padding: '10px' }}>
 {selectedConversaon && (
 <div>
 <h2>Chat with {conversaons.find((conv) => conv.id ===
selectedConversaon).name}</h2>
 <div style={{ maxHeight: '300px', overflowY: 'auto', border:
'1px solid #ccc', padding: '10px' }}>
 {conversaons
 .find((conv) => conv.id === selectedConversaon)
.messages.map((message, index) => (
```

```
 <div key={index} style={{ borderBoom: '1px solid #eee',
paddingBoom: '5px' }}>
 <strong>{message.mestamp}</strong>: {message.text}
</div>
 ))}
 </div>
 <textarea
 rows="3"
 value={newMessage}
 onChange={(e) => setNewMessage(e.target.value)}
/>
 <buon onClick={handleSendMessage}>Send</buon>
</div>
 )}
 </div>
 </div>
 </div>
 );
};

export default MessagingApp;
```

OUTPUT


STUDIO
CHALLENGES
ORGS
MORE
LOGIN
Messaging App
Conversations
Family
Friends
Chat with Family
5:34:48 PM: ok i will call you later good night sweet dreams
5:34:13 PM: hello
Send